University of Nairobi

School of Engineering

DISTRIBUTED TRANSMIT POWER CONTROL IN COGNITIVE RADIO NETWORKS USING WATER-FILLING INTERFACED WITH GAME-THEORETIC LEARNING

BY

Oscar Ondeng

F56/64371/2010

A Thesis submitted in partial fulfillment for the Degree of Master of Science in Electrical and Electronic Engineering, in the Department of Electrical and Information Engineering of the University of Nairobi

November 2015

DECLARATION OF ORIGINALITY

NAME OF STUDENT: Oscar Ondeng

REGISTRATION NUMBER: F56/64371/2010

COLLEGE: Architecture and Engineering

FACULTY/ SCHOOL/ INSTITUTE: Engineering

DEPARTMENT: Electrical & Information Engineering

COURSE NAME: Master of Science in Electrical & Electronic Engineering

TITLE OF WORK: Distributed Transmit Power Control in Cognitive Radio Networks using Water-Filling Interfaced with Game-Theoretic Learning

1) I understand what plagiarism is and I am aware of the university policy in this regard.
2) I declare that this research thesis is my original work and has not been submitted elsewhere for examination, award of a degree or publication. Where other people's work, or my own work has been used, this has properly been acknowledged and referenced in accordance with the University of Nairobi's requirements.
3) I have not sought or used the services of any professional agencies to produce this work
4) I have not allowed, and shall not allow anyone to copy my work with the intention of passing it off as his/her own work
5) I understand that any false claim in respect of this work shall result in disciplinary action, in accordance with University anti- plagiarism policy

Signature: ....

Date:..

Declaration

I, Oscar Ondeng, hereby declare that this thesis is my original work. To the best of my knowledge, the work presented here has not been presented for a degree in any other Institution of Higher Learning.

………………………………………… …………………

Oscar Ondeng Date

This thesis has been submitted for examination with my approval as university supervisor.

………………………………………… …………………

Prof. H. A. Ouma Date

Dedication

To Julia Arwa Ondeng

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to God, who has made all this possible.

I am also grateful to my supervisor, Prof. H. A. Ouma, who has been extremely available and welcoming throughout the duration of the Master's program. He offered many insights and was an encouragement to continued and constant progress, usually giving valuable feedback in a speedy manner.

My heartfelt appreciation also goes to my family, which has been a continued source of motivation and encouragement all through the undertaking of the Master's program. I especially thank my parents, Martin Ondeng and the late Julia Ondeng, who have been insistent on education and inculcated, among others, the necessary virtues of fortitude and industriousness.

ABSTRACT

The electromagnetic spectrum is underutilized despite its physical scarcity. This underutilization is due to inefficient spectrum management techniques. Cognitive radio is a technology that can help achieve flexible spectrum management and thereby increase spectrum efficiency. In this research distributed transmit-power control in cognitive radio networks is studied and represented as a non-cooperative game-theoretic problem. The cognitive radio network is modeled as a single-cell Wide-band Code Division Multiple Access (W-CDMA) network with a number of mobile stations representing the players in the game-theoretic framework. In order to arrive at the Nash Equilibrium (NE) the Iterative Water-Filling (IWF) algorithm is implemented by employing the best response dynamic (BRP). Various characteristics of the NE are investigated such as the convergence speed, the power levels, the signal to interference and noise ratio (SINR) and the utility at NE. Quadrature Phase Shift Keying (QPSK) and Frequency Shift Keying (FSK) are also investigated and compared with respect to the NE and its characteristics. Sequential play is also compared to simultaneous play in the cognitive radio network.

The Nash Equilibrium is found not to be Pareto-optimal and the research presents an iterative algorithm to achieve a strategy that is Pareto-superior to the Nash Equilibrium. Based on the results of the algorithm, a method employing an equation is developed which enables a direct and faster attainment of the Pareto-superior strategy.

In the research a hybrid algorithm that interfaces Iterative Water-Filling and game-theoretic learning is also developed and implemented. The learning component of the game is adaptive and switches between two learning algorithms (Hedge Algorithm and Historical Matching Algorithm) based on the perceived operation of other players in the cognitive radio network. When compared with individual learning algorithms the hybrid-adaptive algorithm yields an improvement in the average utility of 12.65% over Iterative Water-Filling, 9.45% over Historical Matching Algorithm and 67.52% over the Hedge Algorithm. The Iterative Water-Filling component of the hybrid-adaptive algorithm is also seen to offer a convergence that is five times faster than the individual learning algorithms. The novel hybrid-adaptive algorithm offers an improvement on the previous treatments of Iterative Water-Filling and game-theoretic learning. All implementations and simulations are done using MATLAB R2011b.

TABLE OF CONTENTS

ABBREVIATIONS AND ACRONYMS

AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BRP	Best Response to the Previous Play
CA	Communications Authority of Kenya
CEPT	Conference of European Posts and Telecommunications
CR	Cognitive Radio
CRCN	Cognitive Radio Cognitive Network
ECC	European Communications Committee
FCC	Federal Communications Commission
FSK	Frequency Shift Keying
HA	Hedging Algorithm
HMA	Historical Matching Algorithm
IEEE	Institution of Electrical and Electronics Engineers
IWF	Iterative Water-Filling
LTE	Long Term Evolution
MNE	Mixed-strategy Nash Equilibrium
NE	Nash Equilibrium
PNE	Pure-strategy Nash Equilibrium
PU	Primary User
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RKRL	Radio Knowledge Representation Language
RF	Radio Frequency
SDR	Software Defined Radio
SU	Secondary User
SINR	Signal to Interference and Noise Ratio

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 Background

The electromagnetic radio spectrum is a natural resource and its shortage has become more apparent in recent years. This shortage has been due to the physical scarcity of the radio spectrum as well as to the proliferation of wireless devices. However, a deeper analysis of this shortage has revealed that despite the physical scarcity, there is a lot of inefficiency in its utilization. It has been found that some frequency bands in the spectrum are largely unoccupied most of the time whereas some bands are only partially occupied [1]. This state of affairs is illustrated in Figure 1.1 [1], which shows the general nature of spectrum occupancy in an approximately 700 MHz block of spectrum below 1 GHz for three different locations, namely Atlanta, New Orleans and San Diego in the United States. Measurements over various time durations also indicated many time slots in which the spectrum was idle.

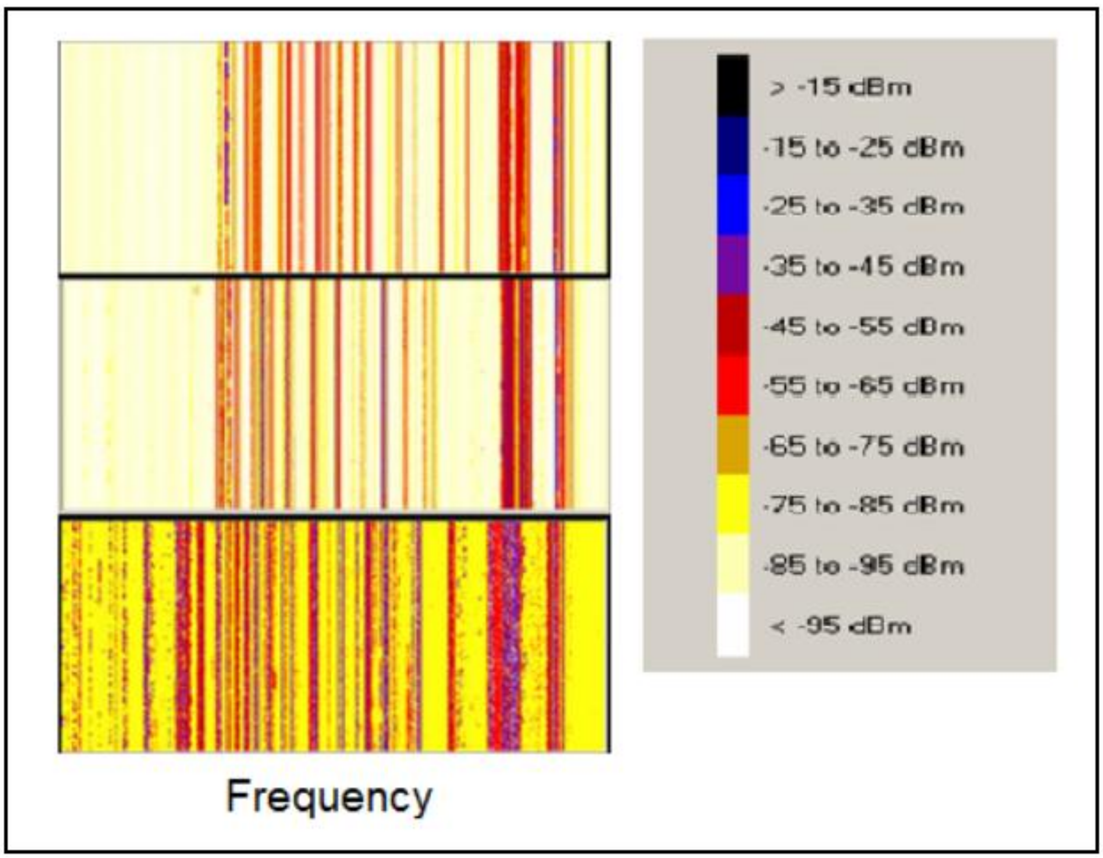


Figure 1.1: Occupancy of a 700 MHz block below 1 GHz

In Kenya, a study done in 2012 by Deloitte & Touche, revealed that of the 436 frequencies allocated 30% were not on air [2]. In this case though the frequencies are assigned, they are totally unutilized. This underutilization of the electromagnetic spectrum gives rise to the concept of spectrum holes, which are bands of frequencies assigned to licensed users, but which, at particular times and specific geographic locations are not utilized by those users [3]. In order to increase the efficiency of the utilization of the spectrum resource, more flexible and dynamic spectrum management techniques and regulations are required.

The current spectrum management policy is a static one in which licensees are assigned frequencies for exclusive use. This is typically regulated by regional bodies such as the Communications Authority of Kenya (CA), the Federal Communications Commission (FCC) in the United States, the Office of Communications (Ofcom) in the United Kingdom, and the Electronic Communications Committee (ECC) of the Conference of European Post and Telecommunications (CEPT) in Europe.

A key technology which can enable the flexible and dynamic spectrum access is cognitive radio. Cognitive radio techniques provide the capability to use or share the spectrum in an opportunistic manner and take advantage of the available spectrum holes and thus increase the efficiency of spectrum utilization. Dynamic spectrum access techniques allow the cognitive radio to operate in the best available channel.

This effort to increase spectrum efficiency becomes important especially due to the scarcity and cost of the spectrum, as well as the need to deploy more services and applications. For instance, there is a general move to migrate television broadcasting to digital broadcast so as to free up TV whitespace. This whitespace could be used for services such as Long Term Evolution (LTE) as well as other services relying on cognitive radio technology.

1.2 Problem Statement

The electromagnetic spectrum is underutilized. This is not so much due to physical scarcity as to inefficient spectrum management techniques and rigid regulations [1]. This gives rise to spectrum holes. There is a need for more flexible and dynamic spectrum management techniques and regulations in order to increase the efficiency of spectrum utilization.

Cognitive radio, first put forward by Mitola [4], was proposed as a novel technique to achieve flexible spectrum management and thereby increase spectrum efficiency [3]. A key task of cognitive radio is the implementation of appropriate dynamic spectrum sharing and allocation techniques and in this, transmit-power control plays an important role. Cooperative mechanisms are often employed to carry out transmit-power control. However, these can often be complicated by competition, especially in a distributed environment. Competition works in opposition to cooperation and can result in negative emergent behaviour, characterized by disorder, chaos and ultimately inefficient spectrum utilization [3].

Game theoretic analysis, coupled with stochastic learning models, is one of the techniques that helps deal with this phenomenon of competition in spectrum sharing in a distributed environment, making stability possible and avoiding exploitation by any of the users. However, these learning models can exhibit a comparatively slow convergence.

Water-filling algorithms, rooted in information theory, offer alternative techniques that can be employed to achieve dynamic spectrum sharing. Water-filling algorithms exhibit fast convergence by virtue of incorporating information on both the transmission channel and the RF environment. However, water-filling lacks learning strategies and, therefore, can leave the users susceptible to exploitation, whereby a user or group of users hog resources and starve the other users.

1.3 Objectives

The overall objective is to develop a model for the implementation of transmit-power control in a cognitive radio network, which has the potential of increasing the efficiency of utilization of the electromagnetic spectrum.

Specifically, the research focuses on algorithms useful for dynamic spectrum sharing and allocation strategies in cognitive radio. The research aims to achieve the following:

i. Model distributed transmit-power control in a cognitive radio network as a non-cooperative game-theoretic problem.
ii. Implement a water-filling algorithm that helps achieve distributed spectrum sharing and allocation in a multiuser scenario.
iii. Implement a learning model to be embedded into the formulated game-theoretic algorithm.
iv. Interface the water-filling approach and the learning model to achieve fast convergence as well as mitigate the exploitation phenomenon.

1.4 Justification of Work

The electromagnetic spectrum is a scarce and expensive resource. As has been pointed out this resource is sometimes underutilized which brings about underutilization. This research seeks to address this problem. Solutions arising from this and similar research can facilitate the availing of spectrum to entities that need to use it. In addition, as TV whitespace becomes available with the migration to digital TV broadcasting it becomes important to investigate schemes that can be used for the utilization of the spectrum that is made available.

1.5 Scope of Work

Cognitive radio systems mainly carry out the tasks of spectrum sensing (radio-scene analysis), spectrum analysis and spectrum decision (transmit-power control and dynamic spectrum management) [3]. These tasks form the cognitive cycle [4].

This work mainly focuses on spectrum transmit-power control (which falls under spectrum decision) implemented based on algorithms relying on game theory, information theory and machine learning. The research does not deal directly with the cognitive tasks of spectrum sensing and spectrum analysis. Prototyping of the algorithms and simulation is carried out using MATLAB.

2 COGNITIVE RADIO

2.1 Cognitive Radio

Cognitive radio is a technology that can help achieve flexible spectrum management and increase the efficient use of the electromagnetic spectrum [3][5]. In a cognitive radio network, a secondary user (unlicensed user) is allowed to access spectrum owned by a primary user (licensed user) so long as the expected Quality of Service (QoS) of the primary user is met and the interference level does not go above a certain threshold.

Akyildiz [5] defined cognitive radio as a radio that can change its transmitter parameters based on interaction with the environment in which it operates. He highlighted cognitive capability, which is the ability of the radio technology to sense or capture information (including temporal and spatial variations) from its radio environment, and re-configurability, which enables the radio to be dynamically programmed according to the radio environment, as the key characteristics of cognitive radio.

Haykin [3] defined cognitive radio as an intelligent wireless communication system that is aware of its surrounding environment and adapts its internal states to statistical variations in the incoming RF stimuli by making corresponding changes in certain operating parameters in real-time, with two primary objectives: highly reliable communications whenever and wherever needed; and efficient utilization of the radio spectrum.

Cognitive radio, therefore, has the ability to exploit its environment to increase spectral efficiency and capacity by exploiting the existence of spectrum holes [6]. Spectrum holes are bands of frequencies assigned to primary users, but which, at a particular time and specific geographic location, are not being utilized by that user [3]. The concept of spectrum holes is illustrated in Figure 2.1 [5]. The spectrum holes can either be temporal or spatial [7]. The more the utilization of spectrum holes the greater the spectrum efficiency.

Spectrum efficiency occurs when the maximum amount of information is transmitted within a given amount of spectrum or equivalently, when the least amount of spectrum is used to transmit a given amount of information [1].

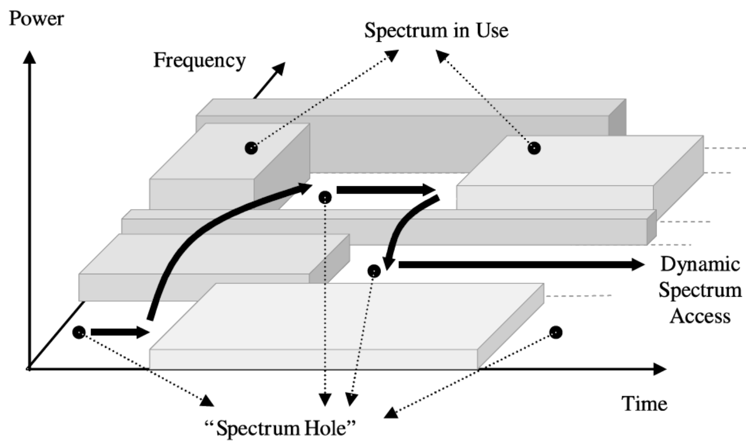


Figure 2.1: Spectrum Holes

Cognitive radio was first proposed by Mitola [4] as an extension to Software-Defined Radio (SDR), an idea which he had earlier proposed. Mitola's concept of cognitive radio was general and included personal services, whose flexibility could be enhanced through a Radio Knowledge Representation Language (RKRL). A key goal was to serve the needs of the user more adequately. He saw Software-Defined Radio as an ideal platform for the realization of cognitive radio. Since his publication in 1999 a lot of research has been done with many advances in this field [7][8]. Part of this has led to the publication of the IEEE 802.22-2011 standard [9], which is a standard for Wireless Regional Area Networks and aims to enable spectrum sharing and broadband wireless access using cognitive radio technology in TV whitespaces between 55 MHz and 862 MHz [10].

2.2 Cognitive Tasks

One way of understanding cognitive radio is by assessing the processes and tasks involved in its realization. These tasks can be viewed in terms of a cognitive cycle, which was first described by Mitola [4]. In illustrating the basic cognitive cycle, Haykin [3] referred to the following fundamental cognitive tasks: radio-scene analysis, channel identification, transmit power control and dynamic spectrum management as illustrated in Figure 2.2 [3]. The cognitive process starts with passive sensing of RF stimuli and culminates with action at the transmitter.

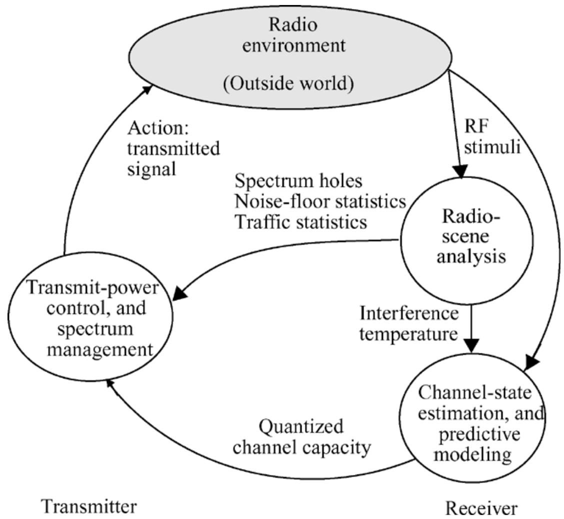


Figure 2.2: The Basic Cognitive Cycle

Akyildiz [5] essentially explained the same cognitive tasks, but employing different terminology. He referred to the basic cognitive functionalities in terms of Spectrum Sensing, Spectrum Analysis and Spectrum Decision.

2.2.1 Spectrum Sensing

Spectrum sensing is also referred to as radio-scene analysis. It is the stage in which unlicensed users – the secondary users (SU) – continuously monitor the activities of the licensed users – the primary users (PU) – to detect the spectrum holes [7]. Haykin [3] indicated that the interference temperature is also estimated at this stage. The interference temperature as a metric to quantify and manage the sources of interference in the radio environment was proposed by the FCC in 2003 [11].

Some of the techniques widely employed for spectrum sensing include energy detection, matched filter detection, cyclostationary feature detection, correlator-based sensing and eigenvalue-based sensing [12].

2.2.2 Spectrum Analysis

During spectrum analysis, the characteristics of the detected spectrum holes are estimated. This entails channel-state information estimation and predictive modeling. The channel capacity, which can be derived from a number of channel parameters is the most important factor for spectrum characterization [5]. It involves capacity estimation based on the interference at the licensed receivers. The channel capacity together with the known threshold for interference help determine the maximum transmit power that the cognitive radio can employ.

2.2.3 Spectrum Decision

During spectrum decision, the cognitive radio determines the frequency and bandwidth of transmission. It also determines the data rate and transmission mode of the communication. This is all based on the information provided from spectrum sensing and analysis. Key tasks in this stage are transmit-power control and dynamic spectrum management.

2.3 Dynamic Spectrum Access: Spectrum Sharing

The SUs determine which channels can be used for transmission as well as when and how to access the channel. A key goal is to protect the PU from undue interference. The SU also needs to take into account the activities of other SUs. In order to achieve a better overall system performance, a sharing strategy can be implemented which entails a dynamic spectrum access [13]. The classification of the sharing strategy can be based on the spectrum access technique, the architecture and the spectrum allocation behaviour [5][13]. These are explained below and illustrated in Figure 2.3 [5].

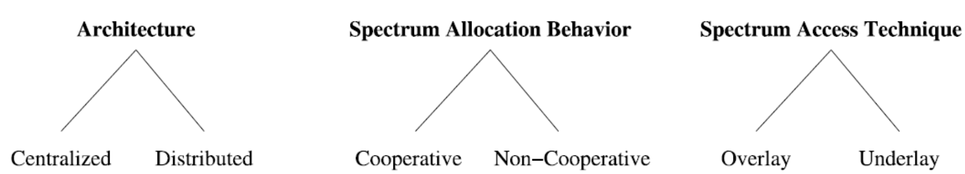


Figure 2.3: Spectrum Sharing Strategies

2.3.1 Spectrum Access Technique

Based on spectrum access technique the spectrum sharing can be open spectrum sharing or licensed spectrum sharing. In open spectrum sharing, all the SUs have equal rights to access the spectrum and freely contend among themselves. In licensed spectrum sharing the PUs have the bands assigned to them and, therefore, have priority over the SUs to use the bands. Licensed spectrum sharing is also referred to as hierarchical spectrum sharing [13]. Hierarchical spectrum sharing is further divided into spectrum underlay and spectrum overlay. In spectrum underlay, the SUs use the spectrum at the same time as the PUs so long as they meet the requirements related to interference and QoS for the PUs. The SUs only transmit when the PUs are not transmitting.

The advantage of open spectrum sharing is that it enables flexible use of the spectrum by a number of users; this can also be done in an *ad hoc* manner. It also facilitates a greater usage of the spectrum which results in a higher efficiency. However, there may be no guarantee that QoS requirements will be met. The advantage of licensed spectrum sharing is that the primary user is guaranteed the usage of the spectrum and a certain QoS can be guaranteed without much difficulty. However, the flexibility in spectrum use is reduced and it can lead to underutilization of the spectrum.

2.3.2 Architecture

Based on the architecture the spectrum sharing can either be centralized spectrum sharing or distributed spectrum sharing [5]. Centralized spectrum sharing entails a central server which is in charge of allocating bands to the other devices [14]. This may, however, involve an increased overhead in terms of signaling. For this reason centralized sharing may be impractical in some cases [13]. Decentralized sharing has been shown in some cases to have benefits over centralized sharing [14].

2.3.3 Spectrum Allocation Behaviour

With regard to spectrum allocation behaviour, the spectrum allocation can be cooperative or non-cooperative. In cooperative spectrum sharing, several SUs work together in order to access the spectrum efficiently. This may be done in a centralized or distributed manner. In the case of non-cooperative spectrum sharing, each SU works to maximize its own benefit, without necessarily taking the global system performance into account. Non-cooperative spectrum sharing is also known as selfish or non-collaborative spectrum sharing. Selfish sharing has a trade-off to be considered since on one hand the non-cooperation may result in a reduced spectrum utilization but on the other hand there is a reduced overhead in communication required among the SUs as seen in cooperative sharing (centralized or decentralized). Therefore, in non-cooperative spectrum sharing the concept of competition arises, in which a particular user may try to exploit the cognitive radio channel for self-enrichment, which in turn, prompts other users to do the same. This results in chaos and inefficient utilization of the spectrum [3].

2.4 Spectrum Allocation and Transmit-Power Control

Transmit-power control is one of the key tasks of the cognitive cycle and plays a big role in carrying out spectrum sharing. It is one of the parameters that needs to be adjusted in order to effect the spectrum sharing and allocation strategy selected. This has to be done in such a way that the interference generated from the SUs is appropriately constrained so as to protect the PUs and to allow as many users as possible to share the spectrum [5]. A number of techniques and methods (spectrum allocation algorithms) can be used to effect transmit-power control and, by extension, the desired spectrum sharing strategy. Cooperative and non-cooperative, as well as centralized and decentralized techniques have been considered and analyzed in [15]. Examples of non-cooperative approaches which have been applied include Game Theory [16][17] and Water-Filling based on information theory [18][19][20]. Spectrum allocation techniques resulting in overlay and underlay sharing strategies have also been assessed in [21].

The spectrum allocation techniques have been gauged and compared using various metrics. The main metrics include interference, spectrum utilization, fairness and throughput [5][14]. The speed of convergence is also useful, and affects these three metrics. The general effort is to optimize these metrics as much as possible.

2.4.1.1 Interference

It is generally desirable that the SUs create limited or no interference to the PUs. Interference between SUs should also be kept at a minimum. Interference is the most common criterion for designing an efficient cognitive spectrum assignment algorithm [22]. The interference temperature was suggested as an approach for quantifying and managing interference [3][11]. The FCC defined the interference temperature as a measure of the RF power generated by undesired emitters plus noise sources that are present in a receiver system per unit of bandwidth. The temperature equivalent of this power is measured in degrees Kelvin (K). The concept of interference temperature is illustrated in Figure 2.4 [11]. It is seen that utilizing the new opportunities for spectrum access raise the noise floor and interference temperature limit. This has the result of reducing the service range.

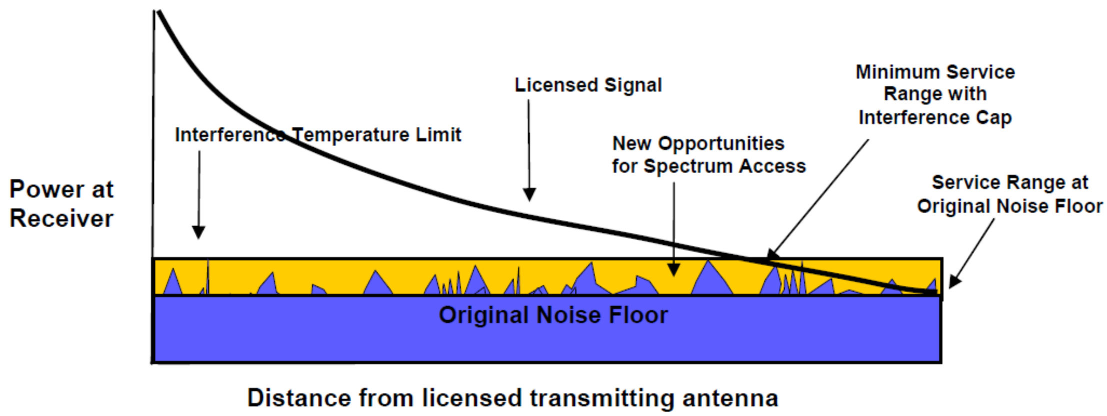


Figure 2.4: Interference Temperature

In 2007, the FCC indicated that the interference temperature was not workable [23]. The National Association for Amateur Radio indicated that the concept was not yet mature [24]. However, it is still relevant in research as the demand for spectrum increases and different players highlight the urgent need for a specification of a metric for measuring harmful interference [25]. Alternative metrics which have been used to characterize interference are Signal to Interference and Noise Ratio (SINR) [16] and outage probability [26], which help safeguard the QoS for the PU. With respect to SINR, a chief aim is to achieve a SINR necessary to maintain a minimum Bit Error Rate (BER).

2.4.1.2 Spectrum efficiency/utilization

The spectrum efficiency can also be expressed as [1]:

$$\text{Spectrum efficiency} = \frac{\text{Information transmitted (I)}}{\text{Spectrum Impacted (U)}} \tag{2.1}$$

The Spectrum Efficiency Working Group [1] indicated the challenges in using efficiency as a metric and stated that the calculation would need to be adjusted to suit specific application. Spectrum utilization as a metric helps maximize either the number of channels assigned to SUs or the number of SUs that are being served in the cognitive network.

2.4.1.3 Throughput

The throughput can be given in terms of the rate of information transmitted [22]. The throughput as a metric helps maximize either the rate of individual users or the total network throughput subject to the maximum transmit power, the link capacity, the maximum interference allowable and the QoS requirements.

2.4.1.4 Fairness

Fairness as a metric aims to achieve an appropriate distribution of spectrum over different users given the QoS requirements and the priorities of the users. Unfairness can arise in cases where one SU selects multiple channels and others are left with no available spectrum (starvation). A possible objective is the maximization of the minimum average throughput per SU. The fairness is akin to the levels of exploitation that are experienced in the network.

2.4.1.5 Speed of convergence

The speed of convergence is pertinent in cases where adaptive or iterative algorithms are employed. It can be assessed by the number of iterations required to reach an equilibrium or stable operation of a system. Closely associated with convergence is the delay, which could either be end-to-end delay or switching delay [22].

2.5 Analyses of Spectrum Sharing based on Spectrum Allocation Behaviour

Peng et al [14] investigated both centralized and distributed approaches from the point of view of utilization and fairness. Peng related that cooperative approaches outperform non-cooperative approaches. In addition the cooperative approaches closely approximate the global optimum. However, in their work the assumption of a static network environment was made.

Zheng and Peng [15] also assessed the effects of collaboration in spectrum access. They also concluded that collaboration yields significant benefits in the utilization and fairness. However, they also only considered a static network environment with a per snapshot optimization. They stated that a dynamic environment could significantly increase the overhead in the case of the cooperative approach.

Nie and Comaniciu [16] made a game theoretical analysis and assessed both cooperative and non-cooperative approaches. They showed that the cooperative case can be modeled as an exact-potential game which converges relatively quickly to a pure strategy Nash Equilibrium (PNE), with a certain degree of fairness and improved throughput. In the non-cooperative approach, a learning algorithm is necessary and results in a mixed strategy Nash Equilibrium (MNE). The fairness is degraded with slightly worse performance. However, in the non-cooperative approach the information exchange necessitated is significantly low. Working with less knowledge about the game is possible and consequently there is less implementation overhead.

Etkin et al [17] assessed spectrum sharing from the point of view of non-cooperative game theory taking into account Gaussian signals. Whereas in a one shot game inefficient solutions may arise, they show that in a repeated or dynamic game the performance loss due to lack of cooperation is small.

Yu and Cioffi [18] showed Iterative Water-Filling to be an efficient solution for optimal resource allocation problems in linear Gaussian Multiple Access and Broadcast Channels. Yu et al [19] considered a multiuser power control problem and modeled it as a non-cooperative game. They implemented a distributive iterative water-filling algorithm without the need for centralized control. They showed that it reached a competitively optimal power allocation and the algorithm was found to give good performance.

Overall in comparing cooperative and non-cooperative spectrum sharing strategies, it is noticed that cooperative settings can result in higher utilization of the spectrum as well as fairness. However, this benefit may not be so high considering the cost of cooperation due to frequent information exchange among users [5]. The performance improvement in the case of cooperative users is acquired at the cost of high environmental knowledge requirement and thereby high signaling overhead [16]. On the other hand, non-cooperative approaches, based for instance on game-theoretic learning or iterative water-filling, could offer the advantage of incomplete information requirement and thereby less signaling overhead. This in turn can give the benefit of less implementation complexity. This could offer potential benefits in the design of cognitive radios for heterogeneous networks [16] in which users are involved in different applications and have various utility functions.

2.5.1 Game-Theoretic Analysis

The transmit-power control in a multiuser cognitive radio environment can be viewed as a game-theoretic problem [3]. Game theory is a field of applied mathematics that describes and analyzes interactive decision situations. It consists of a set of analytical tools that predict the outcome of complex interactions among rational entities, where rationality demands a strict adherence to a strategy based on perceived or measured results [27]. Game theory has extensively been applied to microeconomics but only relatively recently has it been applied to design and analysis of distributed resource allocation algorithms [16].

In the absence of competition, a cooperative game results, which simplifies to an optimal control-theoretic problem and eliminates the game-theoretic aspects of the problem. Game-theoretic analysis of cognitive radio is especially motivated by the concept of a Nash equilibrium (NE), which represents a stable operating point. The NE is a vector of players' actions (an action profile) in which each action is a best response to the actions of all the other players [27].

The notion of the Nash equilibrium, however, has some limitations in that it does not elaborate the underlying process involved in arriving at the equilibrium. Furthermore, it assumes the use of a best response strategy by all the players. However, if one player adopts a non-equilibrium strategy, the response of the other player(s) will also be of a non-equilibrium kind, making the Nash equilibrium inapplicable [3]. To overcome these limitations, learning models may be incorporated into the game theoretic algorithms.

2.5.2 Game-Theoretic Learning

Learning models enable the players to strive for optimality over time. The incorporation of learning is aimed at achieving the result that clever opponents of a player do not exploit dynamic changes or limited resources for their own selfish benefits. It also aims at minimizing bad performance of a player. No-regret algorithms, which are rooted in statistical learning theory, are capable of achieving these aims. The no-regret algorithms are referred to as boosting algorithms by Freund and Shapire [28]. These algorithms result in strong learning models being built around a set of weak learning models. A no-regret algorithm has the advantage that it incorporates a regret agenda such that the learner cannot be deceptively exploited by a cleverer player. However, it converges comparatively slowly [3].

2.5.3 Iterative Water-Filling (IWF)

Water-Filling is a technique rooted in information theory. Given a multiuser cognitive radio environment viewed as a non-cooperative game, the performance of each *unserviced* transceiver is maximized, regardless of what all the other transceivers do, but subject to the constraint that a threshold representing interference is not exceeded. Yu [20] showed that in a Gaussian multiple access channel with multiple transmit and receive antennas, the optimum transmit strategy that maximizes the sum capacity can be found by an iterative water-filling procedure, where each user competitively maximizes its own rate while treating interference from other users as noise. Qi et al [29] developed an enhanced water-filling algorithm which achieved a lower computational complexity than the classical water-filling algorithms.

Iterative Water-Filling has the advantage that it exhibits fast convergence behaviour by virtue of incorporating information on both the channel and the RF environment. However, it lacks the learning strategy that could enable it to guard against exploitation [3].

2.5.4 Learning and Water-filling

There are a number of implementations in the literature of various versions of learning and water-filling as shown in the preceding sections. However, these techniques have been treated in isolation. Haykin [3] pointed out that the performance of iterative water-filling could be improved by interfacing it with a regret-conscious learning algorithm. Ifeh [30] also mentioned the combination of water-filling and learning but his simulation results showed a treatment of the techniques in isolation. His main findings were a comparison of water-filling and no-regret learning giving the strengths and weaknesses of these techniques taken separately, as is illustrated in Table 2.1.

Table 2.1: Advantages and disadvantages of IWF and no-regret learning algorithms [30]

	Iterative Water-filling	**No regret learning**
Strengths	• Converges rapidly • Low computational complexity • Well suited to distributed implementation • Avoids communication links between users	• Guarantees learners cannot be exploited
Weaknesses	• Its sub-optimal • Cannot overcome the behaviour of greedy users	• Slow convergence • High computational complexity

3 GAME THEORY

3.1 Game Theory Fundamentals

Game theory is a bag of analytical tools designed to help us understand the phenomena that we observe when decision-makers interact [31]. Mackenzie [27] defined Game theory as a field of applied mathematics that describes and analyzes interactive decision situations. It consists of a set of analytical tools that predict the outcome of complex interactions among rational entities, where rationality demands a strict adherence to a strategy based on perceived or measured results. Felegyhazi and Hubaux [32] looked at game theory as a discipline aimed at modeling situations in which decision makers have to make specific actions that have mutual, possibly conflicting consequences. Neel [33] first defined a game and then proceeded to define game theory. He defined a game as a model of an interactive decision process. An interactive decision process is a process whose outcome is a function of the inputs of several decision makers who may have potentially conflicting objectives with regard to the outcome of the process. Game theory, then, is a collection of models (games) and analytic tools used to study interactive decision processes [33].

Game theory has extensively been applied to microeconomics and only relatively recently has it been applied to design and analysis of distributed resource allocation algorithms [16]. Other fields in which it has found application include politics, biology, networking, military strategy and wireless communications. Game theory allows for the modeling of scenarios in which there is no centralized entity with a full picture of network conditions [27].

3.1.1 Basic Elements of a Game

The following are the basic elements of a game:

i. Players: these are the decision making entities in the interactive decision process [33]. Games only consider situations where there are two or more players since a single player game would by definition not be an interactive process. The players have interests, which are potentially conflicting. The players are also considered to be rational, meaning that they generally try to maximize their payoffs (utilities) [32] and pursue well-defined exogenous objectives [31]. They also reason strategically, meaning that they take into account their knowledge or expectations of other decision-makers' behaviour [31].

ii. A set of actions or strategies of the players: the strategy of each player can be a single move or a set of moves during the game [32]. Through their strategies the players try to maximize their payoffs.

iii. A set of utilities: these are the players' objective or payoff functions. The utilities express in a compact manner the preference relations of the players [33].

3.1.2 Strategic Games [Static Games]

A strategic game is a model of interactive decision-making in which each decision-maker choses his plan of action once and for all, and these choices are made simultaneously [31]. They are also referred to as single-stage games [32]. A common interpretation of a strategic game is that it is a model of an event that occurs only once [31]. The model consists of:

- Players, represented by the finite set $\boldsymbol{N}$.
- A strategy $s_i \in \boldsymbol{S_i}$ for each player $i \in \boldsymbol{N}$: this represents the set of actions available to player *i*. The collective strategies of all players except player *i* are denoted by $\boldsymbol{s_{-i}}$
- A utility u_i for each player *i*: this is the payoff for each player. **U** is the set of utility functions for all players

The strategy profile **s** is the vector containing the strategies of all players. It is given by

$$\boldsymbol{s} = (s_i)_{i \in \boldsymbol{N}} \tag{3.1}$$

$\boldsymbol{S_i}$ represents the pure individual strategy space of player *i*, meaning that the strategy assigns zero probability to all strategies except one. The joint strategy space of all players is the Cartesian product of all the individual strategy spaces. It is given by

$$\boldsymbol{S} = \prod_{i \in \boldsymbol{N}} \boldsymbol{S_i} \tag{3.2}$$

The utility u_i is a function that characterizes each player's sensitivity to the actions of the other players. It quantifies the outcome of the game for player *i* given the strategy profile **s**. It is a scalar-valued function of the strategy profile:

$$u_i(\boldsymbol{s}) : \boldsymbol{S} \rightarrow \boldsymbol{R} \tag{3.3}$$

Formally, a function that assigns a numerical value to the elements of the action set **S** (i.e. $u(\boldsymbol{s}) : \boldsymbol{S} \rightarrow \boldsymbol{R}$) is a utility function if for all $x, y \in \boldsymbol{S}$, x is at least as preferred as compared to y if and only if $u(x) > u(y)$ [34].

Therefore, based on these basic elements, a game in strategic form (or normal form) is represented as follows:

$$\Gamma = [\boldsymbol{N}, \boldsymbol{S}, \boldsymbol{U}] \tag{3.4}$$

If each player knows the game, Γ i.e. each player knows the set of players, the strategy space and the utility functions, then the game is said to be one with complete information [32]. The strategies in the games can either be pure strategies or mixed strategies. A pure strategy is one in which the strategy assigns a probability of zero to all moves except one in a player's strategy space, $\boldsymbol{S_i}$**.** This means that the players clearly decide on one move or another. For a mixed strategy the players can choose different moves with different probabilities. The mixed strategy $\sigma_i(s_i)$ of a player, therefore, is a probability distribution over his pure strategies $s_i \in \boldsymbol{S_i}$ [32]**.** The mixed strategy space of player *i* is denoted by Σ_i where $\sigma_i \in \Sigma_i$.

3.1.3 Matrix Representation

Strategic games can often be represented in the form of a matrix. For instance, a finite strategic game in which there are two players can be represented as in Figure 3.1.

		p_2	
		L	R
p_1	T	w_1, w_2	x_1, x_2
	B	y_1, y_2	z_1, z_2

Figure 3.1: Matrix representation of strategic games

One player's actions are represented by the rows and those of the other player by the columns. In this case p_1 is Player One and p_2 is Player Two. The set of actions of p_1 are *{T, B}* and those of p_2

are *{L, R}*. The numbers in each of the cells represent the payoffs, the first number being the payoff of p_1, and the second number being the payoff of p_2. For example the payoff from the outcome *{B, R}* is z_1 for p_1 and z_2 for p_2.

3.2 Solving a Game

Solving a game entails determining the likely outcome of the game, given that the players are rational and they are involved in strategic play. It means predicting the strategy of each player, considering the information the game offers and assuming that all the players are rational.

The simplest way of solving a game is relying on dominance [32]. This entails the iterative elimination of dominated strategies and a number of games can be solved in this way. It essentially means iteratively ruling out the strategies that a rational player would not choose based on the payoff functions. The strategy s'_i of player *i* is said to be strictly dominated by his strategy s_i if,

$$u_i(s'_i, \boldsymbol{s_{-i}}) < u_i(s_i, \boldsymbol{s_{-i}}), \forall \boldsymbol{s_{-i}} \in \boldsymbol{S_i} \tag{3.5}$$

3.2.1 Nash Equilibrium

A number of games cannot be solved by iterated dominance techniques. Such games can, however, be solved by making use of the concept of a Nash Equilibrium (NE). A Nash Equilibrium is a joint strategy where no player can increase his utility by unilaterally deviating [27]. Thus, a pure strategy profile **s*** constitutes a Nash Equilibrium if, for each player *i*,

$$u_i(s_i^*, \boldsymbol{s^*_{-i}}) \geq u_i(s_i, \boldsymbol{s^*_{-i}}), \forall s_i \in \boldsymbol{S_i} \tag{3.6}$$

Alternatively, a Nash Equilibrium is a strategy profile comprising of mutual best responses of the players [32]. The best response $br_i(s_{-i})$ of a player *i* to the profile of strategies s_{-i} is a strategy s_i such that:

$$br_i(\boldsymbol{s_{-i}}) = \arg max_{s_i \in \boldsymbol{S_i}} u_i(s_i, \boldsymbol{s_{-i}}) \tag{3.7}$$

When only considering pure strategies, some games may not have a Nash Equilibrium. These games, however, will always at least have a Nash Equilibrium in mixed strategy. Every finite strategic form game has a Nash Equilibrium in either mixed (MNE) or pure (PNE) strategies [27]. In the case of mixed strategies, a mixed strategy profile $\boldsymbol{\sigma} \in \boldsymbol{\Sigma}$ is a Nash Equilibrium if

$$u_i(\sigma_i) \geq u_i(s_i, \boldsymbol{\sigma_{-i}}), \forall\, i \in \boldsymbol{N}, \forall\, s_i \in \boldsymbol{S_i} \tag{3.8}$$

Solving a game, therefore, entails investigating the existence of a Nash Equilibrium and finding out whether it is unique or not. In some games there can exist more than one equilibrium point. The efficiency of the equilibrium points is also studied and this eventually helps in Equilibrium Selection.

3.2.1.1 Arriving at the Nash Equilibrium: Iterative Water-Filling (IWF)

In regard to IWF Yu et al [19] considered a multiuser power control problem in a frequency-selective interference channel, modeled as a non-cooperative game. The multiuser environment he considered was the Digital Subscriber Line (DSL). The IWF technique that he proposed entailed a power allocation scheme that was able to jointly optimize the performance of multiple DSL modems in the presence of mutual interference. The optimization technique aimed at finding the competitive equilibrium (Nash Equilibrium) for the rate maximization game in the DSL, modeled as a frequency-selective Gaussian interference channel. Yu came up with the concept of competitive optimality, in which the Nash Equilibrium can be reached by an iterative water-filling procedure, where each user successively optimizes his power spectrum while regarding other users' interference as noise.

Shum et al [35] looked at synchronous (simultaneous) and asynchronous (sequential) power updates and derived conditions that guarantee convergence. By exploiting some properties of the water-filling method, Qi and Yang [29] proposed a power increment and power decrement water-filling with much lower computational complexity than traditional water-filling.

Consider the model shown in Figure 3.2 of a two-user interference channel.

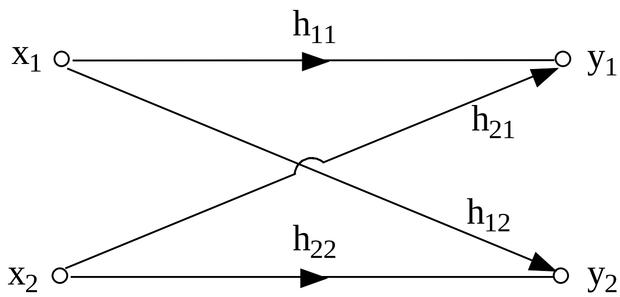


Figure 3.2: A two-user communication channel

where x_1 and x_2 are the transmit signals of transmitter 1 and 2, respectively; y_1 and y_2 are the signals received by receiver 1 and 2, respectively; h_{ij} represents the path gain from user *i* to user *j*. The model can also be represented by

$$y_1 = h_{11}x_1 + h_{21}x_2 + n_1 \tag{3.9}$$

$$y_2 = h_{22}x_2 + h_{21}x_1 + n_2 \tag{3.10}$$

where n_1 and n_2 represent additive noise.

The scenario can be viewed as a non-cooperative game as follows:

- The transmitters and their corresponding receivers are the two players.
- The transmit signals are the pure strategies. These are represented by the transmit power spectra, $P_1(f)$ and $P_2(f)$.
- The utilities are the data rates, R_1 and R_2, of player 1 and player 2, respectively.

Based on Shannon's Gaussian Capacity [36], these data rates are represented as follows [19]:

$$R_1 = \int_0^{F_s} log\left(1 + \frac{P_1(f)|H_{11}(f)|^2}{\Gamma(\sigma_1(f) + P_2(f)|H_{21}(f)|^2)}\right) df \tag{3.11}$$

$$R_2 = \int_0^{F_s} log\left(1 + \frac{P_2(f)|H_{22}(f)|^2}{\Gamma(\sigma_2(f) + P_1(f)|H_{12}(f)|^2)}\right) df \tag{3.12}$$

where $0 \leq f \leq F_s$

$F_s = 1/2T_s$ and T_s is the sampling rate

Γ denotes the SNR-gap

$\sigma_1(f)$ and $\sigma_2(f)$ are the additive white noise at receiver 1 and 2, respectively.

At the Nash Equilibrium, each user's strategy is the optimal response to the other player's strategy. With $P_2(f)$ fixed, the optimal $P_1(f)$ is the solution to the following optimization problem

$$P_1(f) = arg\max_{P_1} R_1 \tag{3.13}$$

$$\text{s.t.} \quad \int_0^{F_s} P_1(f)df \leq \boldsymbol{P_1}$$

$$P_1(f) \geq 0, \quad \forall f$$

Similarly, with $P_1(f)$ fixed, the optimal $P_2(f)$ is the solution to the following optimization problem:

$$P_2(f) = arg\max_{P_2} R_2 \tag{3.14}$$

$$\text{s.t.} \quad \int_0^{F_s} P_2(f)df \leq \boldsymbol{P_2}$$

$$P_2(f) \geq 0, \quad \forall f$$

The solution to this problem gives the water-filling power allocation.

3.2.1.2 IWF Algorithm

The Nash Equilibrium can be reached by an iterative water-filling procedure. IWF works in the following manner: with a fixed total power constraint, each player updates his power allocation by deriving a water-filling power level while regarding all other players' signals as noise. As each player updates his power allocation, the power allocations of the other users are held constant. This forms a single iteration. The procedure is then applied repeatedly until the process converges. In his implementation, Yu considered a target data rate to be achieved. After the water-filling process if a player's data rate is below its target rate, its power is increased, unless this exceeds the power constraint. If, however, a player's data rate is above its target rate, its power is decreased. If the data rate is only slightly above the target rate, its power remains unchanged. Figure 3.3 illustrates the algorithm. The Nash Equilibrium is reached when the desired rate is achieved.

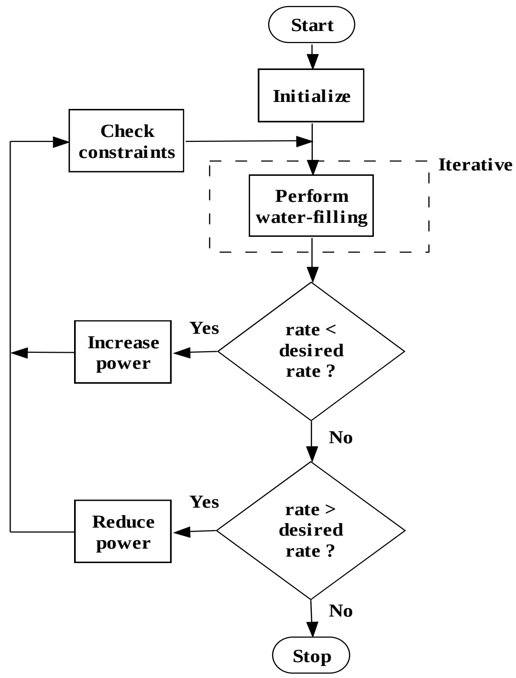


Figure 3.3: Distributed power control based on iterative water-filling

3.2.1.3 Limitations of the Nash Equilibrium

The notion of the Nash equilibrium, however, has some limitations in that it does not elaborate the underlying process involved in arriving at the equilibrium. Furthermore, it assumes the use of a best response strategy by all the players. However, if one player adopts a non-equilibrium strategy, the response of the other player(s) will also be of a non-equilibrium kind, making the Nash equilibrium inapplicable [3].

Furthermore, a conceptual problem arises when there are multiple equilibria. In the absence of an explanation of how players arrive at the same equilibrium, their play may not necessarily correspond to any equilibrium at all. In addition, the hypothesis of exact common knowledge of payoffs and rationality may not apply to many games [37].

To overcome these limitations, learning models may be incorporated into the game theoretic algorithms, enabling the players to strive for optimality over time [37].

3.2.2 Pareto-Optimality

The strategy profile s is Pareto-superior to the strategy profile s' if for any player $i \in \boldsymbol{N}$:

$$u_i(s_i, \boldsymbol{s_{-i}}) \geq u_i(s'_i, \boldsymbol{s'_{-i}}) \tag{3.15}$$

with strict inequality for at least one player [32]. This means that a strategy profile *s* is said to be Pareto-superior to another profile *s'* if the payoff of a player *i* can be increased by changing from *s'* to s without decreasing the payoff of other players. The strategy profile s^{po} is Pareto-optimal (or Pareto-efficient) if there exists no other strategy profile that is Pareto-superior to s^{po}.

3.3 Dynamic Games

These are games in which, rather than making their moves simultaneously without knowledge of the other players' moves, the players make their moves sequentially, meaning that the move of one player is conditioned by the moves of the other players. Repeated games are a subset of dynamic games in which the players interact several times.

Dynamic games are typically represented in extensive form, rather than strategic form though the strategic form can also be used to represent them. In the extensive form the game is represented as a tree. The root of the tree is the start of the game. Each level of the tree is referred to as a stage. The nodes represent the sequence relation of the moves of the players. The players are represented as labels on the nodes. Each node is a complete description of the path preceding it and thus has a unique history. The moves that lead to each node are represented on each branch of the tree. Each terminal node (leaf) defines a potential end of the game (an outcome) and is assigned the corresponding payoff. Figure 3.4 is an extensive form representation of the game depicted in Figure 3.1.

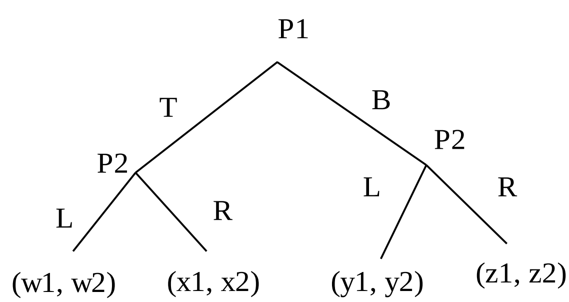


Figure 3.4: Extensive form representation of dynamic games

Dynamic games can either have perfect information or imperfect information. A game with perfect information is one in which all the players have perfect knowledge of all the previous moves in the game at any time they have to make a new move. The games can also be finite-horizon games, in which there exist a finite number of stages, or infinite-horizon games.

In repeated games, players try to maximize their expected payoff over multiple rounds of the game. The concept of the Nash Equilibrium is also pertinent in the case of repeated games. It can be applied in some situations in the form of a sub-game perfect equilibrium.

3.4 Potential Games

A game $\Gamma = [\boldsymbol{N}, \boldsymbol{S}, \boldsymbol{U}]$ is an ordinal potential game [27][38] if there exists a function $V : \boldsymbol{S} \rightarrow \boldsymbol{R}$ such that for all $i \in \boldsymbol{N}$, all $x, z \in \boldsymbol{S_i}$ and all $s_{-i} \in \boldsymbol{S_{-i}}$,

$$u_i(x, s_{-i}) - u_i(z, s_{-i}) > 0 \Leftrightarrow V(x, s_{-i}) - V(z, s_{-i}) > 0 \quad (3.16)$$

The game is an exact potential game if equality holds as shown in equation 3.17.

$$u_i(x, s_{-i}) - u_i(z, s_{-i}) = V(x, s_{-i}) - V(z, s_{-i}) \quad (3.17)$$

Equation 3.16 and 3.17 show that every exact potential game is an ordinal potential game. In both cases, the potential function V (ordinal potential function in equation 3.16 and the exact potential function in equation 3.17) represents the change in utility for any unilaterally deviating player.

Monderer [38] showed that if Γ is a finite ordinal potential game, then Γ has at least one Nash Equilibrium in pure strategies. If Γ is an infinite ordinal potential game with a compact strategy space **S** and a continuous potential function V, then Γ has at least one Nash Equilibrium in pure strategies.

3.5 Markov Games

These are also referred to as stochastic games. In a Markov game, the history at each stage of the game is summarized by a state, and movement from state to state follows a Markov process, meaning that the state of the next round of the game depends on the current state and the current action profile. In addition to the elements contained in the strategic form game, the Markov game is also characterized by state variables $m \in \boldsymbol{M}$ and transition probability $q\ (m^{k+1}\ | m^k, \boldsymbol{a^k})$, which denotes the probability that the state in the next round is m^{k+1} conditional on being in state m^k in round k and on the playing of the action profile $\boldsymbol{a^k}$. A Markov perfect equilibrium is a profile of Markov strategies which yields a Nash Equilibrium in every proper sub game [27].

3.6 Game-Theoretic Learning

The notion of the Nash equilibrium assumes the use of a best response strategy by all the players. However, if one player adopts a non-equilibrium strategy, the response of the other player(s) will also be of a non-equilibrium kind, making the Nash equilibrium inapplicable [3]. A conceptual problem also arises when there are multiple equilibria. In the absence of an explanation of how players arrive at the same equilibrium, their play may not necessarily correspond to any equilibrium at all. In addition, the hypothesis of exact common knowledge of payoffs and rationality may not apply to many games [37].

Learning helps mitigate these limitations. No-regret learning aims at minimizing the cumulative loss relative to the loss suffered by the best strategy [28] as well as maximizing rewards in non-deterministic settings [39].

If Γ is a repeated game and $\mathbf{Q}$ is the joint mixed strategy space made up of the individual mixed strategy spaces $\{\boldsymbol{Q_i}\}_{i\in \boldsymbol{N}}$ of each player, at time t the regret ρ_i player i experiences for playing strategy q_i^t rather than strategy s_i is given by [40]

$$\rho_i(s_i, q_i^t | \boldsymbol{s}^t_{-\boldsymbol{i}}) = u_i(s_i, \boldsymbol{s}^t_{-\boldsymbol{i}}) - u_i(q_i^t, \boldsymbol{s}^t_{-\boldsymbol{i}}) \tag{3.18}$$

where $s_i \in \boldsymbol{S_i}$, $q_i \in \boldsymbol{Q_i}$.and $\boldsymbol{s_{-i}}$ represents the collective strategies of all players except player i.

The mixed strategy of player *i* at time *t + 1* is given by [40]

$$q_i^{t+1} = L_i(h_i^t) \tag{3.19}$$

where L_i is the learning algorithm and h_i^t is the subset of the history known to player *i* at time *t*. Through the learning algorithm a player makes a prediction of the current strategy of the other players and then chooses a strategy in response to that prediction. The more accurate the prediction the better response the player will be able to make. q_i^t represents a probability distribution over the pure strategies of player *i*. The probabilities of the pure strategies can be seen as weights corresponding to the pure strategies and which can be updated during each iteration in a learning process.

3.7 Learning Algorithms

3.7.1 Best Response to the Previous Strategy of the Opponent (BRP)

The iterative water-filling as implemented using the best response to the previous strategy of the opponent is the fundamental algorithm to which the other learning algorithms are compared. In BRP there is no significant history that is stored by the players. Each player only maintains information of the previous stage game. In simultaneous play, the best response correspondence is the situation in which each player plays its optimal strategy in response to the predicted current strategies of the other players based on the previous stage game. The current strategies of the other players in the current stage game are essentially taken to be the same strategies played in the previous stage game.

3.7.2 Regret-Matching Algorithm (RMA)

For the regret-matching algorithm [40][41] the weights for each stage game are updated based on the cumulative regrets accrued from the previous stage games. It therefore needs to keep a history of the previous stage games. The history size can vary from one stage game up to a maximum of all previous stage games. The cumulative regret felt by player *i* for not having played strategy s_i through time *t* is given by

$$R_i^t(s_i) = \sum_{x=1}^{t} \rho_i^x(s_i, s_i^x | s_{-i}^x)$$

$$(3.20)$$

The update rule is:

$$q_i^{t+1}(s_i) = \frac{[R_i^t(s_i)]^+}{\sum_{s_i' \in S_i}[R_i^t(s_i')]^+} \tag{3.21}$$

where $X^+ \overset{\text{def}}{=} max\{X, 0\}$

3.7.3 The Hedge Algorithm (HA)

The Hedge algorithm [28][40] depends on the cumulative utilities achieved in the previous game stages and employs an exponential updating scheme. This algorithm also necessitates maintaining information of the previous stage games.

Let $u_i^t(s_i)$ denote the cumulative utility obtained by user *i* through time *t* by choosing strategy s_i:

$$u_i^t(s_i) = \sum_{x=1}^{t} u_i(s_i, s_{-i}^x) \tag{3.22}$$

The weight (probability) assigned to strategy s_i at time $t + 1$ is given by:

$$q_i^{t+1}(s_i) = \frac{\beta^{u_i^t(s_i)}}{\sum_{s_i' \in S_i} \beta^{u_i^t(s_i')}} \tag{3.23}$$

where β is the hedge parameter [28] and in this case is constrained by $\beta > 1$. The values used for $u_i^t(s_i)$ are normalized.

The Hedge algorithm maintains a weight vector ($\boldsymbol{q}^t$) which is updated during each iteration. The weights are required to be non-negative and for the initial weight vector they sum up to one, so that $\sum_{i=1}^{N} q_i^1 = 1$, where N is the number of players.

3.7.4 The Adapted Weighted Majority Algorithm (AWM)

This is based on the Weighted Majority Algorithm of Littlestone and Warmuth [42]. For the adapted weighted majority algorithm an initial positive weight of one is associated with each strategy of the learning player. The update rule is given by:

$$q_i^{t+1}(s_i) = \begin{cases} \alpha q_i^t(s_i), & s_i = \operatorname*{argmax}_{s_i \in S_i} u_i(s_i, s_{-i}^x) \\ \beta q_i^t(s_i), & otherwise \end{cases} \tag{3.24}$$

where $\alpha > 1$ and $0 < \beta < 1$.

The multiplicative factors α and β have the effect that the weight of the strategies which achieves the highest utility in an iteration for the learning player are increased whereas the weights of the strategies which achieve lower utilities are reduced. Therefore, the strategies producing low utilities are employed less and less whereas the strategies producing higher utilities are employed more and more.

3.7.5 History-Matching Algorithm (HMA)

The History Matching Algorithm [43] essentially searches through the history of stage games of the adversary and looks for a sequence that matches the last few strategies (the pattern) of the adversarial player. If the pattern is found, the algorithm examines the strategy played by the opponent immediately after the position of the occurrence of the pattern and predicts that that strategy will be played again at the current iteration. To effect the history matching a cross-correlation is made of the last few strategies with the history of strategies available. The point at which the maximum correlation occurs is used to predict the next strategy

4 METHOD AND IMPLEMENTATION

4.1 Modeling

In this research the transmit-power control is modeled as a non-cooperative repeated game (infinite horizon), which is a sequence of stage games, each stage game being a normal form game; the players are taken to be myopic. The simulation set-up is for a W-CDMA network containing a single cell and a varying number of users. MATLAB was used to implement the different algorithms and perform the simulations.

The normal form game is completely defined by specifying the following tuple:

$$\Gamma = [\boldsymbol{N}, \boldsymbol{P}, \boldsymbol{U}] \tag{4.1}$$

Where $\boldsymbol{N}$ is the set of players, $\boldsymbol{P}$ is the joint strategy space made up of the individual strategy spaces $\{\boldsymbol{P_i}\}_{i\in\boldsymbol{N}}$ and $\boldsymbol{U}$ is the set of utility functions $\{u_{\boldsymbol{i}}\}_{i\in\boldsymbol{N}}$. In this case $\boldsymbol{N}$ consists of the mobile stations in the network and varies (2 or more users). $\boldsymbol{P}$ consists of the possible transmission powers of the mobile stations and has a minimum of 0 and a maximum of 2 W, based on the power limits of a class-1 mobile station of a W-CDMA network. The class-1 mobile stations are among the most widely used of the hand-held devices. Note that equation 4.1 uses the symbol $\boldsymbol{P}$ as compared to $\boldsymbol{S}$ used in equation 3.4 since in this simulation the strategies consist in power levels.

4.1.1 The Utility Function

The utility function helps in the modeling of a power-control game. In the game the cognitive radios adjust their power levels in order to maximize their utility, local as well as global. In doing this, the radios aim at balancing the signal to interference and noise ratio (SINR) and throughput against the power consumption. The SINR is a measure of the quality of signal reception for the wireless user. Typically, a user would like to achieve a high quality reception while at the same time expending as little energy as possible [34]. A typical scenario which can be used to model this setting is that of a single-cell mobile network in which the mobiles adapt their transmit-power levels. Studies looking at the power control problem as a game have been conducted by [34][44] [45] [46]. These authors have used similar utility functions which help maximize the SINR and thereby achieve a certain minimum Quality of Service (QoS). Some of them have also introduced pricing to increase the Pareto efficiency of the solution that is arrived at. The utility function for user i is specified as follows [34]:

$$u_i(p_i, \boldsymbol{p}_{-i}) = \frac{LR}{Mp_i}(1 - 2 \times BER)^M \tag{4.2}$$

where *L* is the number of information bits in each frame

M is the total number of bits in each frame

R is the transmission rate (bits/second)

p_i is the transmission power (watts)

BER is the Bit Error Rate

The units of the utility function are bits per joule. Therefore, the utility is a measure of the amount of information that can be transmitted per joule of energy. For instance, in the case of a mobile device, a higher utility would mean that the device can transmit more information for the period during which the battery has charge. The utility function can be set to represent any given modulation technique [34] by choosing the appropriate BER

4.1.2 Network Details

For a W-CDMA network with a spreading factor of 256, each frame has the following parameters: L = 100, M = 150 (assuming 1/3- rate coding), R = 15kbps. The modulation scheme used in W-CDMA is QPSK and the bit error rate for player i is given by [47]:

$$BER_{QPSK} = \frac{1}{2}\left(erfc\left(\sqrt{\gamma_i}\right) - \frac{1}{4}\, erfc^2\left(\sqrt{\gamma_i}\right)\right) \tag{4.3}$$

where γ_i is the signal to interference and noise ratio (SINR) of the ith user. The SINR of user i is given by [44]:

$$\gamma_i = \frac{W}{R}\frac{p_i h_i}{\sum_{j=1, j\neq i}^{N} p_j h_j + \sigma^2}$$

(4.4)

where W = 5 MHz (this is the W-CDMA bandwidth)

$\sigma^2 = 2 \times 10^{-14}\, W$ (AWGN at the receiver)

h_i is the path gain from user i to the base station.

For a calculation of σ^2 see Appendix A. As can be seen from equation (4.4), the SINR at the receiver depends on all the users. Therefore, there is an interdependence among the users, which is aptly expressed by the utility as given in equation (4.2), which depends on γ_i through the BER. This emphasizes the appropriateness of applying game theory in such a problem. In some instances, FSK has been used for purposes of comparison. The bit error rate for FSK is given by [47]:

$$BER_{FSK} = \frac{1}{2} e^{-\gamma_i/2} \tag{4.5}$$

For the path gains, the Extended Hata Model (COST-231) [48] is employed. The basic formula for the propagation loss in dB given by the Extended Hata Model is

$$L_{XHata} = 46.33 + (44.9 - 6.55 \log h_1) \log d_{km} + 33.9 \log f_{MHz} - a(h_2) - 13.82 \log h_1 + C \tag{4.6}$$

Where h_1 and h_2 are the base station and mobile antenna heights in meters, respectively. According to COST-231 h_1 can range from 30m – 200m and h_2 can range from 1m – 10m. d_{km} is the link distance in kilometers and can range from 1km – 20km; f_{MHz} is the centre frequency in megahertz and ranges from 1500MHz – 2000MHz; $a(h_2)$ is the antenna height-gain correction factor and is given by

$$a(h_2) = (1.1 \log f_{MHz} - 0.7)\ h_2 - (1.56 \log f_{MHz} - 0.8) \tag{4.7}$$

The parameters used for this study are: h_1 = 30 m (the base station average height), h_2 = 1.5 m (the mobile station average height), f_{MHz} = 1900 MHz; d_{km} are random distances from 1 km to 2 km. For a small – medium city or a suburban area, C = 0 [48] (C represents a correction factor introduced in the Extended Hata Model to improve on the accuracy of the model).

4.2 Convergence to Nash Equilibrium using Iterative Water-Filling (IWF)

In this research the iterative water-filling [3][19] [35] technique is implemented in an attempt to converge to the Nash Equilibrium (NE). The NE is desirable because it represents a stable operating point. The IWF is implemented by employing the Cournot Adjustment Process [37], which is essentially a best response correspondence [27] where each player's strategy is the best response to the other players' strategies. The flowchart of Figure 4.1 illustrates the best response correspondence (BRP).

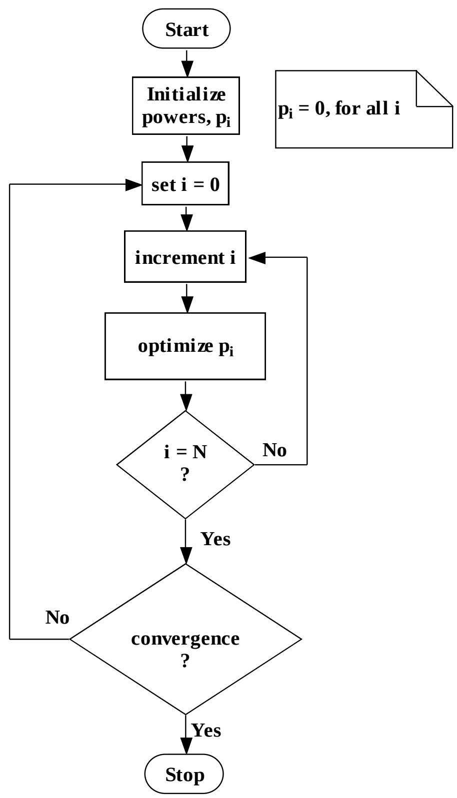


Figure 4.1: IWF using the best response correspondence

The step "optimize p_i" for sequential play is carried out as follows:

$$p_i^{t+1} = \underset{p_i}{\operatorname{argmax}}\, u_i(p_i, p_{-i}^*) \tag{4.8}$$

$$\text{s.t.} \qquad 0 \le p_i \le p_{max}$$

where $p_{-i}^* = \begin{cases} p_{-i}^{t+1}, \; if\ current\ strategy\ of\ opponent\ is\ known \\ p_{-i}^{t}, \; if\ current\ strategy\ of\ opponent\ is\ not\ known \end{cases}$

p_{max} = 2 W

4.3 Algorithm for a Pareto-Superior Power Vector

Goodman [45] illustrated that for the power control game, a NE reached may not be Pareto-optimal. This is illustrated by the fact that if all the powers of all users are simultaneously reduced by a factor μ then a power vector can be found which is Pareto-superior to the NE, such that

$$p_i^{ps} = \mu p_i, \ \forall \, i \in N \tag{4.9}$$

where p_i^{ps} is the pareto superior power vector.

A Pareto superior outcome to the NE can be achieved using the following iterative algorithm:

a. Play the power control game and adjust the powers of the players until the NE is reached.
b. Determine the value of μ_{peak} as follows:
   i. Reduce μ in small steps from one towards zero.
   ii. At each step multiply the power vector at NE by μ (a scalar) and calculate the individual utilities as well as the sum utility.
   iii. Repeat the reduction of μ until a point at which the sum utility decreases instead of increasing or until a point at which the utility for at least one player decreases.
   iv. Take the value of μ at that point to be μ_{peak}.
c. Adjust the power vector at NE for all players by multiplying each power by the factor μ_{peak}.

$$p_i^{ps} = \mu_{peak} p_i, \ \forall \, i \in N \tag{4.10}$$

4.4 Implementation of Different Learning Algorithms

In the research a number of learning algorithms (cf. Section 3.6.1) were also implemented and compared in a two-player setting. The main metric was the utility derived by the players. The details of the setting were as follows:

a. The game is a simultaneous game of imperfect information.
b. The players are at equal distances from the base station so that none of them is disadvantaged by virtue of position. A player nearer the base station than another would have an advantage in that it would expend less power to transmit the same amount of information.

c. There are two players, one employing a learning algorithm, and the second one employing an adversarial strategy (deterministic and/or non-deterministic).
d. The player employing the learning uses the different learning algorithms implemented in turn against the adversarial player. The utilities derived when using the different learning algorithms were compared with each other as well as with the best response dynamic.
e. The players employ pure strategies as opposed to mixed strategies.
f. Play first proceeds to reach the NE point, and then the adversarial player breaks away from the NE.
    i. The adversarial player in one instance plays with a deterministic strategy and in another instance with a non-deterministic strategy.
    ii. The learning player employs the learning algorithm in an attempt to learn the strategy of the adversarial opponent over 50 learning iterations. 50 is chosen so as to give time to the learner to learn the strategy of the adversarial player.
g. The players have a reduced strategy set. The deterministic and non-deterministic strategies of the adversarial player are as follows:
    i. For the deterministic strategies, the adversarial player repeatedly sets his power in each stage game to the following levels in turn: 0.5 W, 1 W, 1.5 W and 1.9 W.
    ii. For the non-deterministic strategies, the adversarial players randomly sets his power in each stage game to the same levels in as in a) above with each level having a probability of occurrence of 0.25. Other probability distributions are possible; it will be the goal of the learning player to learn the probability distribution of the adversarial player, whichever it may be, and adjust its own strategy accordingly.

The adversarial player in these simulations represents a player who does not necessarily play according to the best response dynamic. The adversarial player can be viewed as an embodiment of an entire wireless network with many players. It can thus represent the sum total of conditions and changes that may occur in the wireless network and which are perceived by a learning player trying to maximize its utility.

4.5 Interfacing Iterative Water-Filling and Game-Theoretic Learning

Iterative water-filling has the advantage that it exhibits fast convergence behaviour by virtue of incorporating information on both the channel and the RF environment. However, it lacks the learning strategy that could enable it to guard against exploitation [3]. Game-theoretic learning models on the other hand have the advantage that they incorporate a regret agenda such that the learner cannot be deceptively exploited by a cleverer player. However, they converge comparatively slowly [3].

In this research iterative water-filling is combined with game-theoretic learning models to come up with an algorithm which draws from the advantages of both techniques. The hybrid is intended to exhibit fast convergence behaviour as well as incorporate a regret agenda which helps reduce the possibility of exploitation by a cleverer or malicious player.

4.6 Proposed Hybrid-Adaptive Algorithm

Figure 4.2 shows the algorithm employed to integrate IWF and no-regret game-theoretic learning. The play begins with the first stage game. At this point the iterative water-filling algorithm is first run. IWF has been chosen to run first because it has a fast convergence characteristic and it is desirable that the game converges to the Nash Equilibrium as soon as possible. Each time the IWF algorithm is run for all players. The IWF algorithm is run until the NE is reached. A condition is put in place to ensure that this iteration does not create an infinite loop in the event that a NE is not reached or does not exist. This condition consists in checking that the iterations do not exceed a predetermined maximum number of iterations. Note that the existence of the NE is partly dependent on the utility function chosen. In the modeling for this research, the utility function employed has been shown to result in the existence of the NE (cf. Section 4.1.1).

Once the NE is reached, a switch is made in order to run the game-theoretic learning algorithms. The learning algorithms have a slower convergence characteristic but guard well against possible exploitation or adaptations in the network conditions. The learning algorithms are used to maintain the NE (after convergence) and to reduce the possibility of exploitation, a task which the learning algorithms performs better than the IWF. The two learning algorithms used were HMA and HA. These two algorithms were selected based on their superior empirical performance.

The parameters relevant to the learning algorithms employed are first set. These include the NE window length, the variation threshold, the correlation threshold and the pattern length. The NE window length is a moving window containing a history of the strategies of the players once the NE is reached for a number of iteration. A longer window implies more iterations making up the history for analysis.

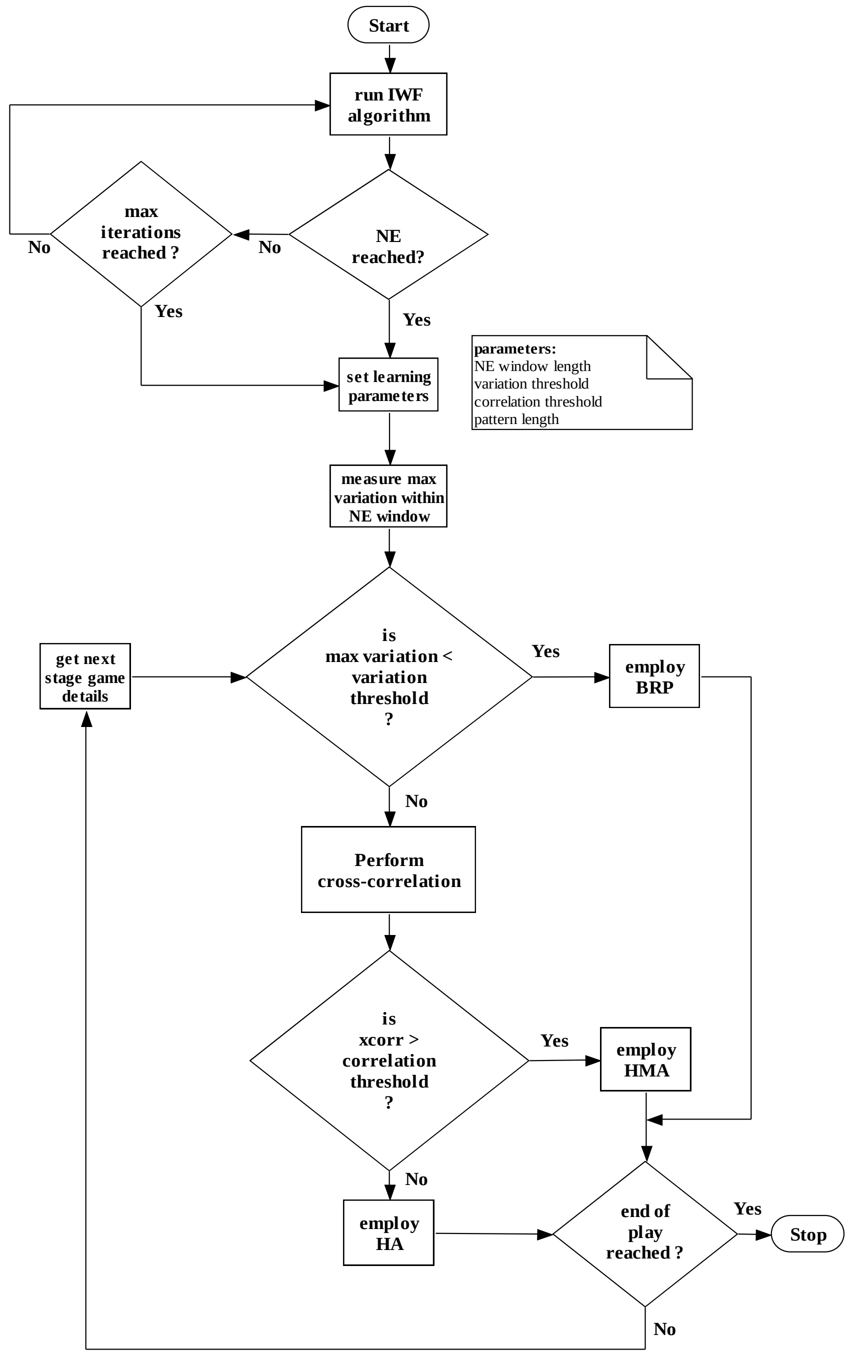


Figure 4.2: Hybrid-Adaptive Algorithm

Exceeding the variation threshold prompts the use of either HMA or HA; if the variation threshold is not exceeded BRP is used. The variation in this case is a measure of the magnitude of changes occurring in power levels between successive iterations.

The pattern length is the number of previous iterations taken into account in order to perform a correlation with the rest of the history of strategies available. If the correlation threshold is exceeded HMA is employed, otherwise HA is employed.

The adaptive learning algorithm draws on the strengths of two learning algorithms and enables the learning player to detect the mode of play of the adversarial player and adapt its own mode of play to suit that of the adversarial player. This is done in an attempt to maximize the utility accrued to the learning player. Thus, when the learning algorithms are being utilized, there is an adaptation from one learning algorithm to another, based mainly on the measured variation of power levels and the value for the correlation performed, which are used to detect the mode of play of the opponent.

Therefore, the overall hybrid-adaptive algorithm used in the research entails a hybrid of Iterative Water-Filling and learning, with the learning component consisting of an algorithm that adapts between two different learning algorithms, HMA and HA.

5 RESULTS AND DISCUSSION

5.1 Convergence to Nash Equilibrium (NE) using Iterative Water-Filling

The number of players is denoted by N, with $N \geq 2$. Each of the players play sequentially using the best response dynamic over a number of iterations in an attempt to converge to a Nash Equilibrium. The sequential play is one with perfect information in that the player whose turn it is to play has knowledge of the most recent strategies employed by all other players. For the utility function being used, a pure strategy Nash Equilibrium has been shown to exist [34]. Each player chooses the best response to the actions of the other players. Therefore, the Nash Equilibrium consists of the mutual best responses of all the players in a given game.

Table 5.1 – Table 5.5 show some of the convergence characteristics for 2, 3, 4, 5 and 10 players all employing QPSK.

Table 5.1: NE for 2 players

Player	distance from BS (km)	NE Tx Power (W)	NE SINR (dB)	NE Utility (10^6 b/J)
1	1.1319	0.0050	5.2761	1.6804
2	1.2476	0.0070	5.2422	1.1925
Iterations to reach NE: 2			**Sum Utility**	**2.8729**

Table 5.2: NE for 3 players

Player	distance from BS (km)	NE Tx Power (W)	NE SINR (dB)	NE Utility (10^6 b/J)
1	1.1319	0.0051	5.2939	1.6529
2	1.2476	0.0072	5.3047	1.1731
3	1.3304	0.0090	5.2875	0.9355
Iterations to reach NE: 2			**Sum Utility**	**3.7616**

Table 5.3: NE for 4 players

Player	distance from BS (km)	NE Tx Power (W)	NE SINR (dB)	NE Utility (10^6 b/J)
1	1.1319	0.0052	5.3103	1.6260
2	1.2476	0.0073	5.2909	1.1541
3	1.3304	0.0091	5.2590	0.9203
4	1.4815	0.0133	5.2620	0.6300
Iterations to reach NE: 2			**Sum Utility**	**4.3305**

Table 5.4: NE for 5 players

Player	distance from BS (km)	NE Tx Power (W)	NE SINR (dB)	NE Utility (10^6 b/J)
1	1.1319	0.0052	5.2223	1.5990
2	1.2476	0.0074	5.2757	1.1353
3	1.3304	0.0093	5.2874	0.9053
4	1.4815	0.0135	5.2539	0.6197
5	1.5569	0.0161	5.2607	0.5203
Iterations to reach NE: 2			**Sum Utility**	**4.7797**

Table 5.5: NE for 10 players

Player	distance from BS (km)	NE Tx Power (W)	NE SINR (dB)	NE Utility (10^6 b/J)
1	1.1319	0.0057	5.2414	1.4642
2	1.2476	0.0081	5.2873	1.0394
3	1.3304	0.0101	5.2569	0.8288
4	1.4815	0.0148	5.2738	0.5674
5	1.5569	0.0176	5.2653	0.4764
6	1.6306	0.0207	5.2616	0.4048
7	1.7246	0.0253	5.2790	0.3323
8	1.8104	0.0300	5.2756	0.2800
9	1.8927	0.0350	5.2625	0.2394
10	1.9611	0.0397	5.2676	0.2113
Iterations to reach NE: 3			**Sum Utility**	**5.8442**

In all cases the play of the game (iterations) converges to a Nash Equilibrium (NE) with NE SINR for all players being close to the expected SINR (cf. Appendix A). The close SINR of all players gives an indication as to the fairness of the NE from the point of view of the signal received at the base station. The play converges to the same NE regardless of the order in which the players play. Generally, the higher the number of players the higher the number of iterations needed to converge to the NE.

Figure 5.1 shows the number of iterations taken to reach convergence for QPSK. In each iteration each player adjusts his power to maximize utility and the iterations continue until an equilibrium is reached. Figure 5.2 shows the convergence in the case of FSK.

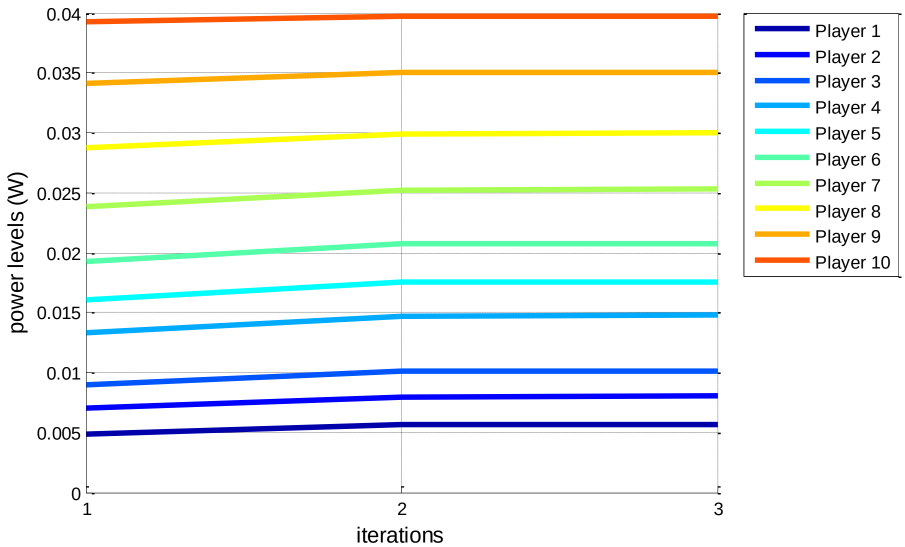


Figure 5.1: Convergence to NE of 10 players using QPSK

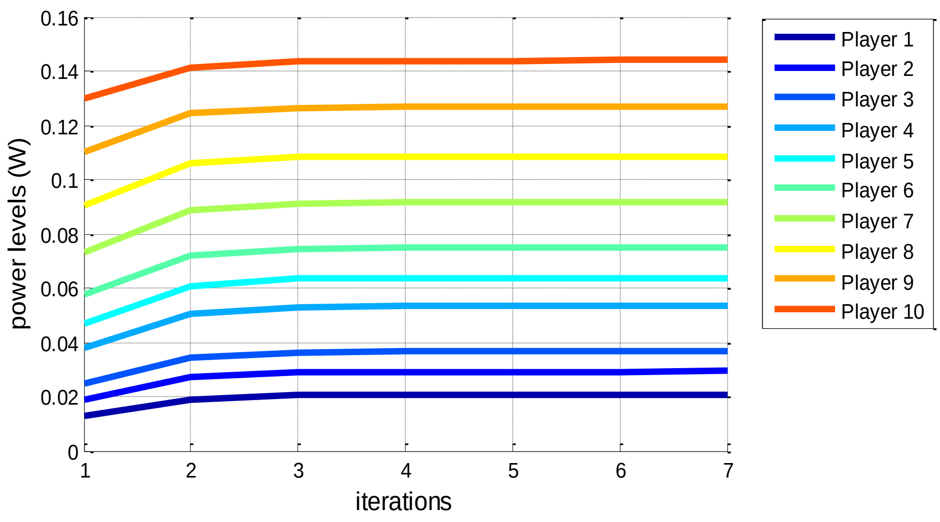


Figure 5.2: Convergence to NE of 10 players using FSK

When compared to the utility function that employs FSK, it is noticed that QPSK takes fewer iterations to converge. This can be attributed to the fact that QPSK a different BER function to arrive at the utility is used. In the case of FSK it takes 7 iterations to converge. The characteristics of the NE that the FSK situation converges to are shown in Table 5.6.

Table 5.6: NE for 10 players using FSK

Player	**distance from BS (km)**	**NE Tx Power (W)**	**NE SINR (dB)**	**NE Utility (10^5 b/J)**
1	1.1319	0.0208	13.9272	4.1716
2	1.2476	0.0293	13.9245	2.9608
3	1.3304	0.0367	13.9077	2.3610
4	1.4815	0.0536	13.9054	1.6163
5	1.5569	0.0638	13.8959	1.3570
6	1.6306	0.0751	13.8980	1.1530
7	1.7246	0.0915	13.8993	0.9464
8	1.8104	0.1086	13.9038	0.7976
9	1.8927	0.1270	13.9026	0.6820
10	1.9611	0.1439	13.9008	0.6018
			Sum Utility	**16.6475**

5.2 Utility as a function of Power

The variation of utility with power shown in Figure 5.3 illustrates the characteristics of the utility function.

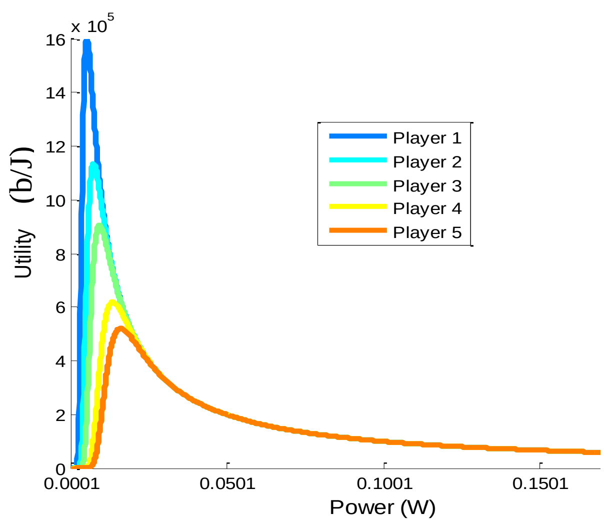


Figure 5.3: Variation of utility with power for 5 players using QPSK in the last iteration

As power for each user increases the utility increases but only until a certain point where an increase in power results in a reduced utility. Therefore, a user cannot indiscriminately increase power in an attempt to increase utility. The players closer to the base station achieve a higher utility than those farther away. This is expected since the mobile stations closer to the base station generally have to expend less power to achieve a target SINR.

5.3 Pareto Optimality

Goodman [45] illustrated that for the power control game, a NE reached may not be Pareto optimal. This is illustrated by the fact that if all the powers of all users are simultaneously reduced by a factor μ (equation 5.1) then a power vector can be found which is Pareto superior to the NE, such that

$$p_{i_{new}} = \mu p_i, \ \forall \ i \ \in N \tag{5.1}$$

For the QPSK simulation, the parameter μ was varied from 0 to 2 in steps of 0.001 for games with up to 60 players. Table 5.7 shows the variation of μ_{peak} with the number of players, N, who are all at a distance of 1.2 km from the base station so as to present a fairer situation without any players being advantaged by virtue of position. μ_{peak} is the value of μ at which the peak in the sum utility occurs. It was noticed that the peak in the sum utility does not occur at the same value of μ for different numbers of players. For higher numbers of players it was observed that peak of the utility sum occurred at lower values of μ.

The Pareto-superior outcomes (at μ_{peak}) in Table 5.7 were effectively obtained by employing the algorithm for Pareto-superior power vector in Section 4.3. The Algorithm for Pareto-improvement helps achieve a higher overall utility for the entire system while guaranteeing that the utility for all players is at least equal to their utilities at NE. The utilities and power vectors at μ_{peak} do not constitute an equilibrium point and would therefore need to be enforced. The enforcing mechanism can be done via a minimum of cooperation or implementation of punishment in the repeated game [32].

Table 5.7: Variation of μ_{peak}, sum utilities and average utilities with the number of players

N	μ_{peak}	Sum Utility at μ_{peak} (10^6 b/J)	Average Utility per Player at μ_{peak} (10^6 b/J)	Sum Utility at NE (10^6 b/J)	Sum Utility at NE (10^6 b/J)
2	0.96	2.7331	1.3666	2.7331	1.3666
3	0.96	4.0280	1.3427	4.0280	1.3427
4	0.96	5.2715	1.3179	5.2715	1.3179
5	0.96	6.4612	1.2922	6.4612	1.2922
6	0.92	7.6847	1.2808	7.5774	1.2629
7	0.93	8.8146	1.2592	8.7269	1.2467
8	0.94	9.9019	1.2377	9.8369	1.2296
9	0.95	10.9470	1.2163	10.9050	1.2117
10	0.95	11.9410	1.1941	11.9290	1.1929
11	0.96	12.9060	1.1733	12.9060	1.1733
12	0.96	13.8340	1.1528	13.8340	1.1528
13	0.96	14.7110	1.1316	14.7110	1.1316
14	0.9	15.5530	1.1109	15.2350	1.0882
15	0.91	16.3520	1.0901	16.0970	1.0731
16	0.92	17.1120	1.0695	16.9170	1.0573
17	0.93	17.8300	1.0488	17.6920	1.0407
18	0.95	18.5110	1.0284	18.4220	1.0234
19	0.95	19.1480	1.0078	19.1040	1.0055
20	0.96	19.7360	0.9868	19.7360	0.9868
21	0.88	20.3220	0.9677	19.8280	0.9442
22	0.89	20.8490	0.9477	20.4520	0.9296
23	0.91	21.3390	0.9278	21.0330	0.9145
24	0.92	21.7940	0.9081	21.5680	0.8987
25	0.93	22.2110	0.8884	22.0570	0.8823
26	0.85	22.5930	0.8690	21.8380	0.8399
27	0.87	22.9390	0.8496	22.3130	0.8264
28	0.88	23.2520	0.8304	22.7450	0.8123
29	0.89	23.5300	0.8114	23.1330	0.7977
30	0.82	23.7740	0.7925	22.7150	0.7572
31	0.84	23.9860	0.7737	23.0820	0.7446
32	0.85	24.1650	0.7552	23.4090	0.7315

N	μ_{peak}	Sum Utility at μ_{peak} (10^6 b/J)	Average Utility per Player at μ_{peak} (10^6 b/J)	Sum Utility at NE (10^6 b/J)	Sum Utility at NE (10^6 b/J)
33	0.79	24.3120	0.7367	22.8640	0.6928
34	0.81	24.4270	0.7184	23.1640	0.6813
35	0.76	24.5130	0.7004	22.5470	0.6442
36	0.77	24.5690	0.6825	22.8140	0.6337
37	0.78	24.5960	0.6648	23.0440	0.6228
38	0.74	24.5930	0.6472	22.3990	0.5894
39	0.75	24.5640	0.6298	22.5940	0.5793
40	0.71	24.5070	0.6127	21.9390	0.5485
41	0.68	24.4230	0.5957	21.2910	0.5193
42	0.69	24.3150	0.5789	21.4450	0.5106
43	0.66	24.1820	0.5624	20.8080	0.4839
44	0.63	24.0250	0.5460	20.1930	0.4589
45	0.61	23.8450	0.5299	19.6030	0.4356
46	0.59	23.6420	0.5140	19.0390	0.4139
47	0.57	23.4190	0.4983	18.4980	0.3936
48	0.55	23.1740	0.4828	17.9810	0.3746
49	0.51	22.9090	0.4675	16.9520	0.3460
50	0.5	22.6280	0.4526	16.5060	0.3301
51	0.47	22.3270	0.4378	15.6230	0.3063
52	0.45	22.0110	0.4233	14.8250	0.2851
53	0.41	21.6770	0.4090	13.7470	0.2594
54	0.38	21.3280	0.3950	12.8100	0.2372
55	0.36	20.9680	0.3812	11.9900	0.2180
56	0.32	20.5920	0.3677	10.8190	0.1932
57	0.29	20.2060	0.3545	9.6774	0.1698
58	0.25	19.8080	0.3415	8.4781	0.1462
59	0.22	19.3990	0.3288	7.3371	0.1244
60	0.18	18.9830	0.3164	6.0328	0.1005

Figure 5.4 illustrates the variation of sum utility with μ in the specific case of 40 players. When μ = 1, the scenario is that of the NE. Values of μ greater than 1 result in reduced utilities. However, when μ reduces slightly, there is an increase in the sum utility without a reduction in utility for any one of the players; as μ reduces further beyond μ_{peak} the utility begins to drop rapidly. It is seen that some strategies exist that are Pareto-superior to the NE. The algorithm for Pareto-superior power vector given in Section 4.3 can thus be employed to improve on the NE. In the case of 40 players μ_{peak} was found to be 0.71 as shown in Figure 5.4.

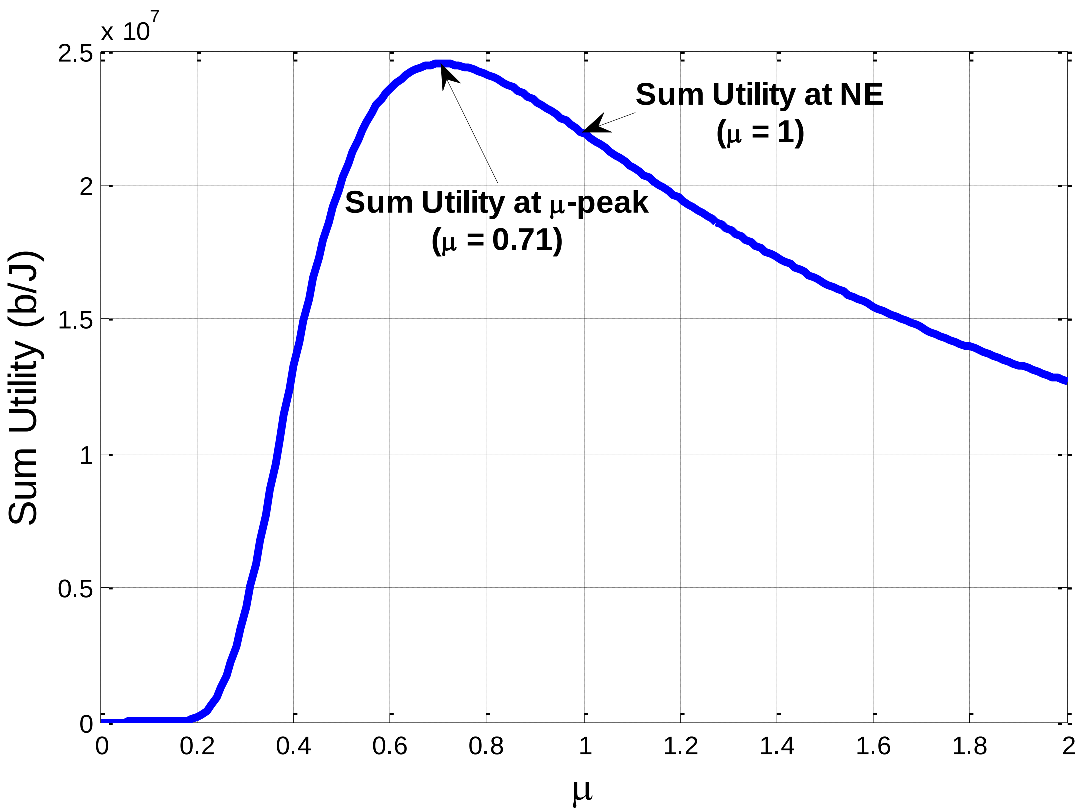


Figure 5.4: Sum of utility against μ for 40 players

The variation of Table 5.7 is illustrated in Figure 5.5. It is noticed that the Sum Utility at NE (μ = 1) and at μ_{peak} generally increases as the number of players increase. As the players increase further, the Sum Utility levels off and begins to drop. This points to the fact that increasing the number of players does not indefinitely increase the sum utility that can be drawn from the network as congestion always leads to poorer throughput. It is noticed that the average utility per player (both at NE and at μ_{peak} reduces steadily as the number of players increases, which is expected, since more players means that the same electromagnetic spectrum has to be shared by more users.

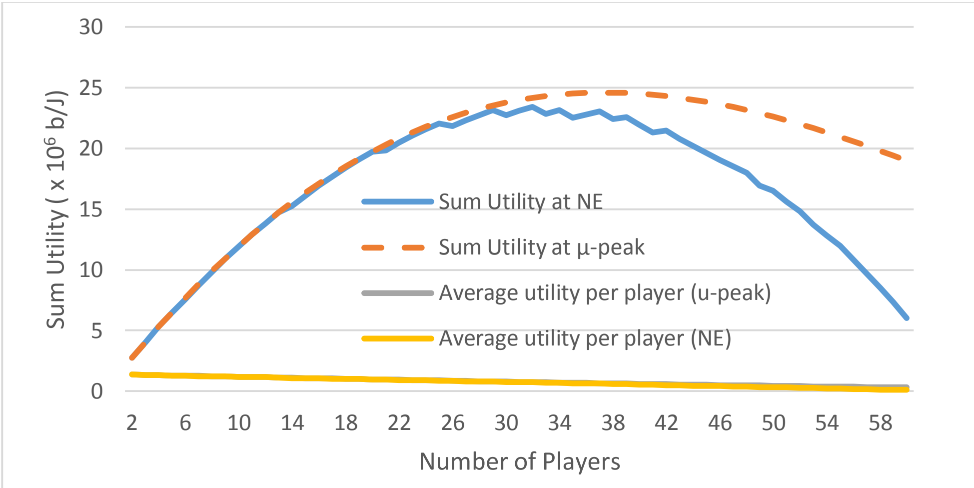


Figure 5.5: Variation of Sum Utility with number of players

Figure 5.6 illustrates graphically the variation of μ_{peak} with the number of players. It is noticed that as the number of players increases the μ_{peak} reduces.

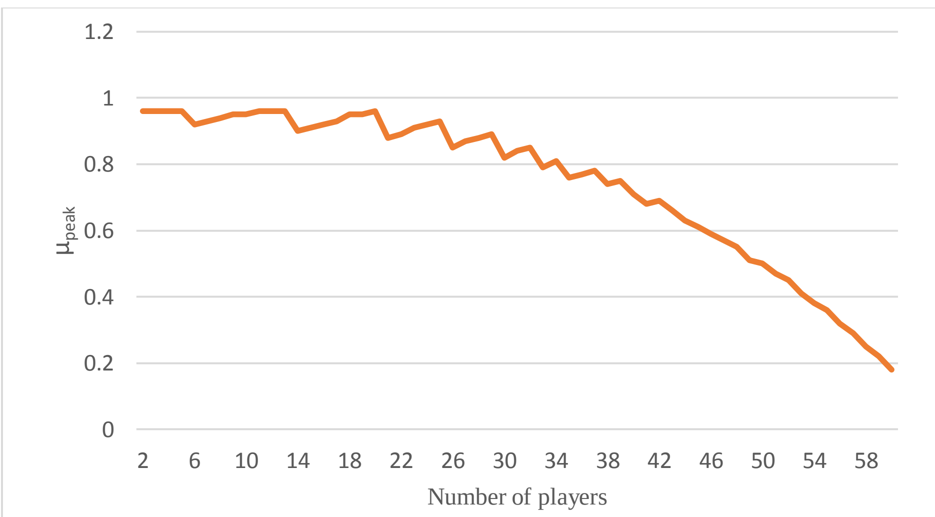


Figure 5.6: Variation of μ_{peak} with number of players

Figure 5.7 shows the individual utilities achieved for 60 players at μ_{peak} ($\mu = 0.18$) as well as at NE ($\mu = 1$). Figure 5.8 shows the corresponding power levels of the players at μ_{peak}. The value of μ_{peak}

is arrived at using the algorithm for a Pareto-superior power vector. In both figures, the 60 players are placed at equal intervals between 1 km and 2 km such that player 1 is 1km from the base station and player 60 is 2 km from the base station. Figure 5.7 and Figure 5.8 illustrate the fact that the iterative algorithm for Pareto-improvement yields higher utilities.

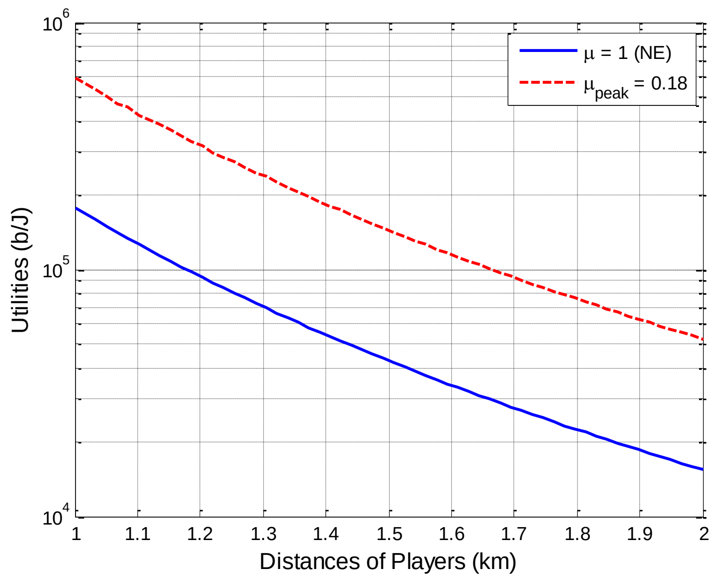


Figure 5.7: Utilities for 60 players with $\mu = 1$ (NE) and $\mu = \mu_{peak}$ (0.18)

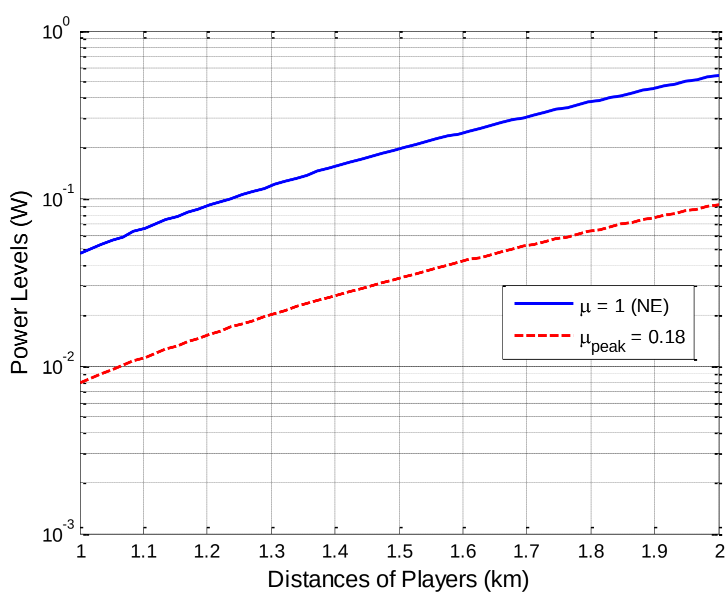


Figure 5.8: Power levels for 60 players with $\mu = 1$ (NE) and $\mu = \mu_{peak}$ (0.18)

Determination of μ_{peak}:

Based on the data of Figure 5.6 a curve-fitting procedure was used to establish a relationship between μ_{peak} and the number of players. Following from this, the value of μ_{peak} can be expressed as

$$\mu_{peak} = (-0.0301N^3 - 0.4412N^2 + 0.0128N + 9481) \times 10^{-4} \tag{5.2}$$

where N is the number of players.

Figure 5.9 shows a comparison of the variation of μ_{peak} with the number of players based on equation 5.2 and the experimental variation of the number of players with μ_{peak}.

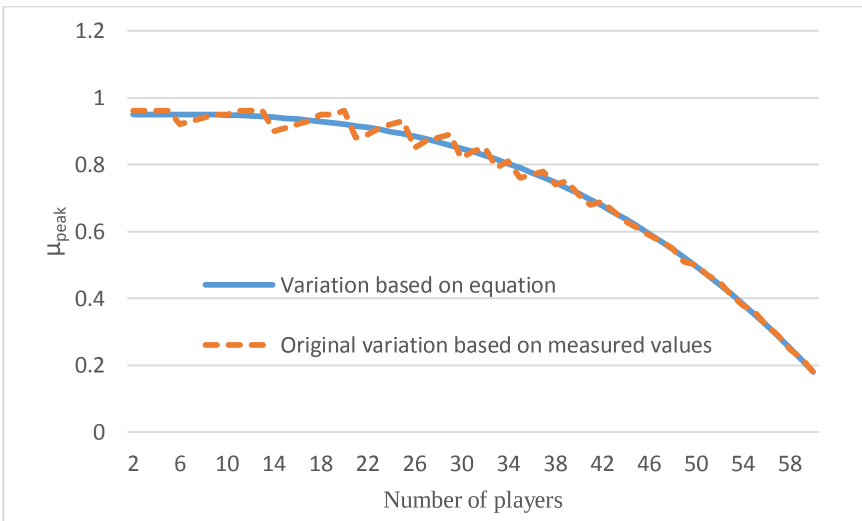


Figure 5.9: Comparison of values of μ_{peak} based on measured values and developed equation

To assess equation 5.2, utilities and power levels for 60 players, at varying distances from the base station, were acquired by improving on the NE using the value of μ_{peak} acquired via equation 5.2. In Figure 5.10 and Figure 5.11 these utilities and powers, respectively, are compared with those acquired using the algorithm for Pareto-improvement which were illustrated in Figure 5.7 and Figure 5.8. Figure 5.10 and Figure 5.11 show that the use of the equation results in a similar Pareto-improvement as in the case of the Algorithm for Pareto-Improvement. The use of the equation has the advantage of faster execution as opposed to the use of the algorithm, which is iterative and therefore takes more time to execute.

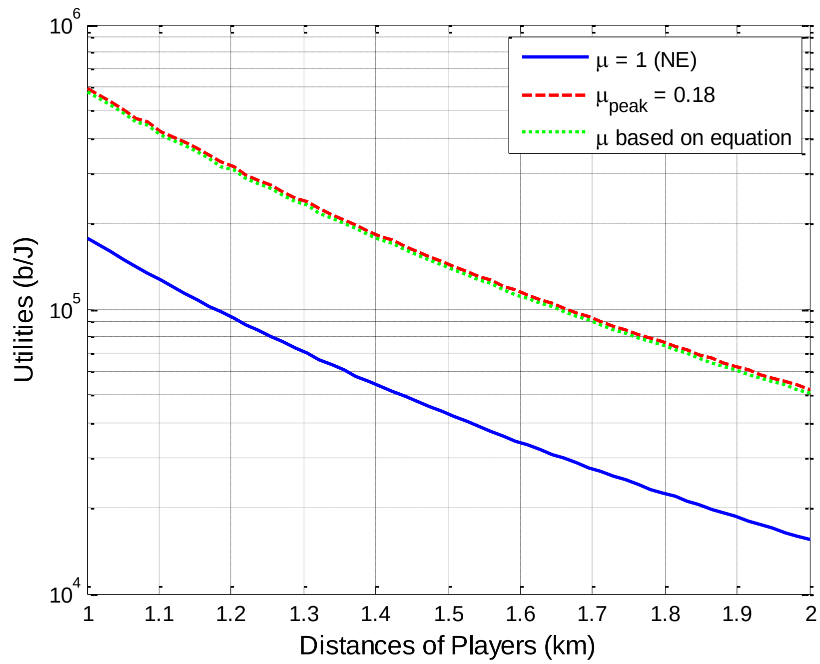


Figure 5.10: Utilities for 60 players using different values of μ

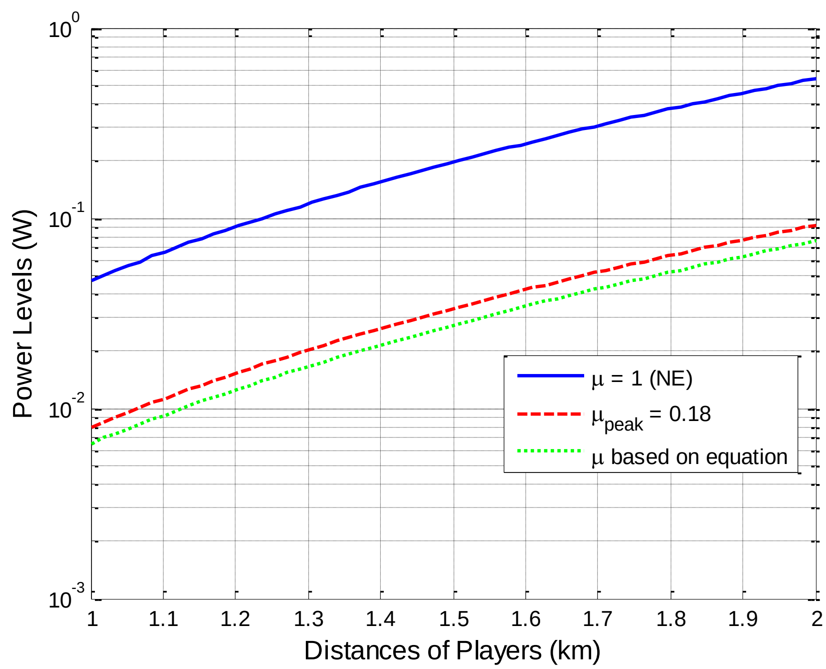


Figure 5.11: Powers for 60 players using different values of μ

This technique for finding a Pareto-superior power vector to the NE offers an improvement to the iterative algorithm for Pareto-superior power vector presented in Section 4.3 as well as to other methods such as the method of pricing employed by [34][46] in that a direct equation can be employed to arrive at μ_{peak} which can then be used to improve on the NE. The technique used in [34][46] entails iteratively looking for the parameter, which can result in a slower process.

5.4 Convergence to NE for Players with different Utility Functions

A simulation was done for 5 players using the utility function based on QPSK and 5 players for the utility function based on FSK. It was found that the play still converged to NE (Figure 5.12); however, the NE SINR's were different. For Figure 5.12, the distances from base stations in km were as indicated in Table 5.8. It is noted that for the players based on QPSK the NE SINR was close to 5.3 whereas for those based on FSK the NE SINR was close to 13.9; these are the expected NE SINR for QPSK and FSK, respectively.

Table 5.8: NE details for a combination of QPSK and FSK

Player	Modulation Scheme	Distance from BS (km)	NE Tx Power (W)	SINR (dB)	NE Utility (10^5 b/J)
1	QPSK	1.1319	0.0067	5.2682	1.2521
2	FSK	1.2476	0.0243	13.9075	0.3566
3	QPSK	1.3304	0.0118	5.2512	0.7087
4	FSK	1.4815	0.0445	13.9037	0.1947
5	QPSK	1.5569	0.0206	5.2694	0.4073
6	FSK	1.6306	0.0624	13.9079	0.1389
7	QPSK	1.7246	0.0295	5.2627	0.2841
8	FSK	1.8104	0.0902	13.9081	0.0961
9	QPSK	1.8927	0.0410	5.2710	0.2047
10	FSK	1.9611	0.1195	13.9028	0.0725
				Sum Utility	**3.7156**

Iterations to reach NE: 5

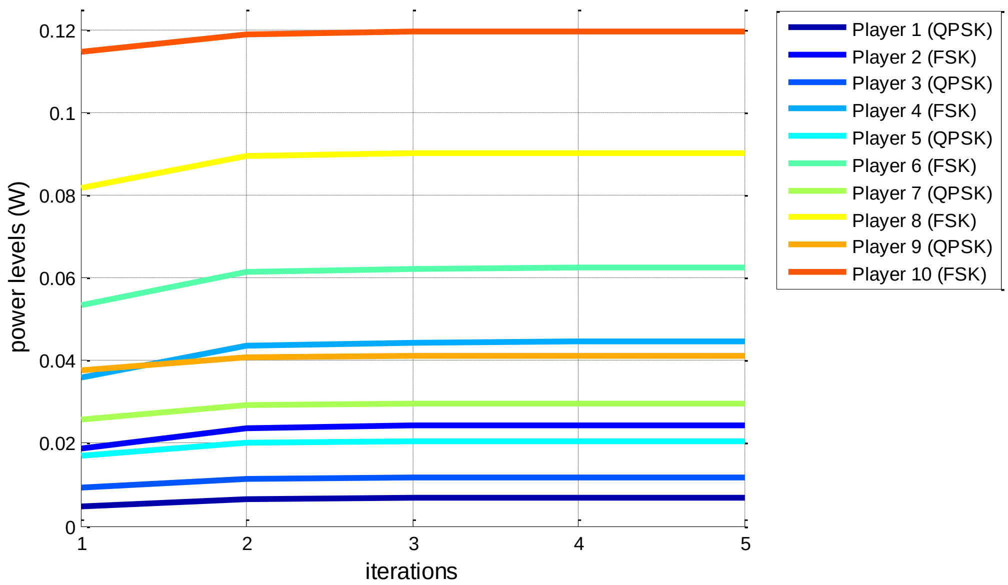


Figure 5.12: Convergence for 10 players employing different utility functions

5.5 Nash Equilibrium in Simultaneous Play

The previous simulations were performed based on sequential repeated play (Cournot Adjustment) where players have perfect knowledge of the strategies (including the most recent) and utility functions of the other players. In simultaneous repeated play with imperfect information the players have knowledge of the history of the game but do not know the most recent strategies of the other players until the current stage game is played. All the players make their moves simultaneously. This represents a network environment in which the different transceivers may communicate simultaneously. For simultaneous play the step "optimize p_i" as illustrated in Figure 4.1 is carried out as follows:

$$p_i^{t+1} = \underset{p_i}{\operatorname{argmax}}\, u_i(p_i, p_{-i}^t) \tag{5.3}$$

$$\text{s.t.} \qquad 0 \leq p_i \leq p_{max}$$

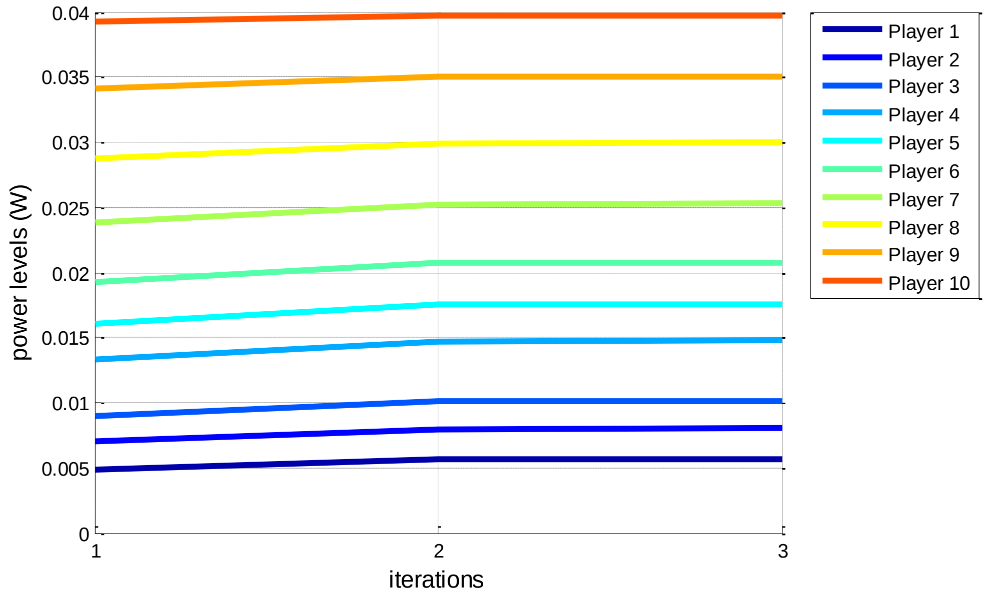
This section presents the results for the simultaneous play and compares them with the results for the sequential play. Table 5.9 shows the details of the NE for 10 players in simultaneous play.

Table 5.9: NE for 10 players in simultaneous play

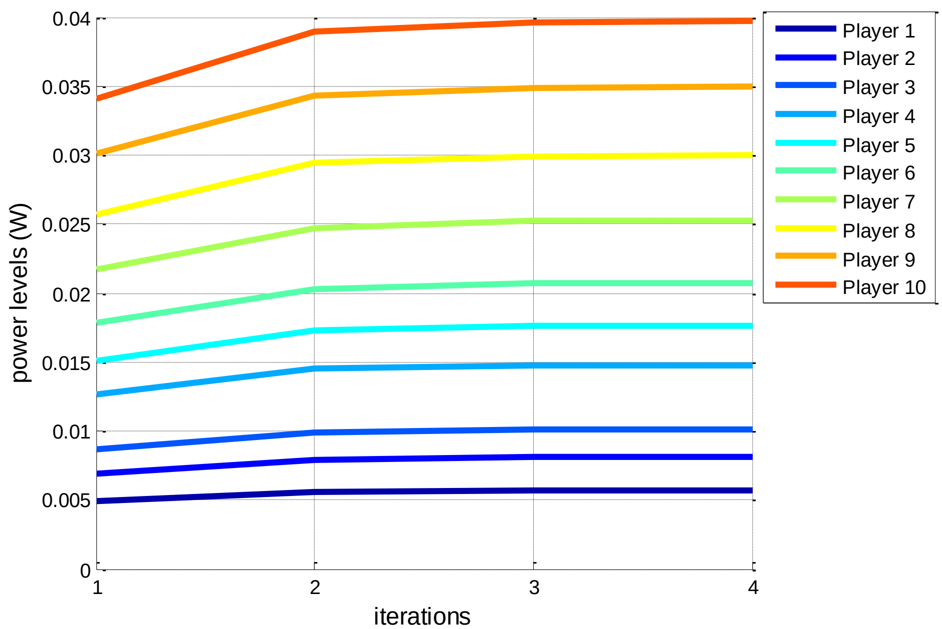
Player	distance from BS (km)	NE Tx Power (W)	NE SINR (dB)	NE Utility (10^6 b/J)
1	1.1319	0.0057	5.2414	1.4642
2	1.2476	0.0081	5.2873	1.0394
3	1.3304	0.0101	5.2569	0.8288
4	1.4815	0.0148	5.2738	0.5674
5	1.5569	0.0176	5.2653	0.4764
6	1.6306	0.0207	5.2616	0.4048
7	1.7246	0.0253	5.2790	0.3323
8	1.8104	0.0300	5.2756	0.2800
9	1.8927	0.0350	5.2625	0.2394
10	1.9611	0.0397	5.2676	0.2113
Iterations to reach NE: 4			**Sum Utility**	**5.8442**

Figure 5.13 and Figure 5.14 show the convergence to NE in the case of 10 players and 20 players, respectively, using QPSK. Figure 5.13(a) and Figure 5.14(a) are the case of sequential play with perfect information (Cournot Adjustment) whereas Figure 5.13(b) and Figure 5.14(b) are the result of simultaneous play with imperfect information. It is noted that the sequential play and the simultaneous play converge to the same equilibrium points i.e. the play converges to the same transmit powers, SINR and utility for all players in the case of simultaneous play as compared to sequential play. This is also seen from Table 5.5 and Table 5.9.

However, it is noted that the simultaneous play takes more iterations to reach the NE point. In the case of 10 players it took 4 iterations as compared to 3 for sequential play to reach convergence. In the case of 20 players it took 8 iterations to reach convergence with simultaneous play as compared to 5 iteration for sequential play. Sequential play converges faster due to the fact that the players have knowledge of the moves of other players in any individual stage game whereas in simultaneous they don’t have the knowledge since the play is taking place at the same time. Having this extra information results in a faster convergence.

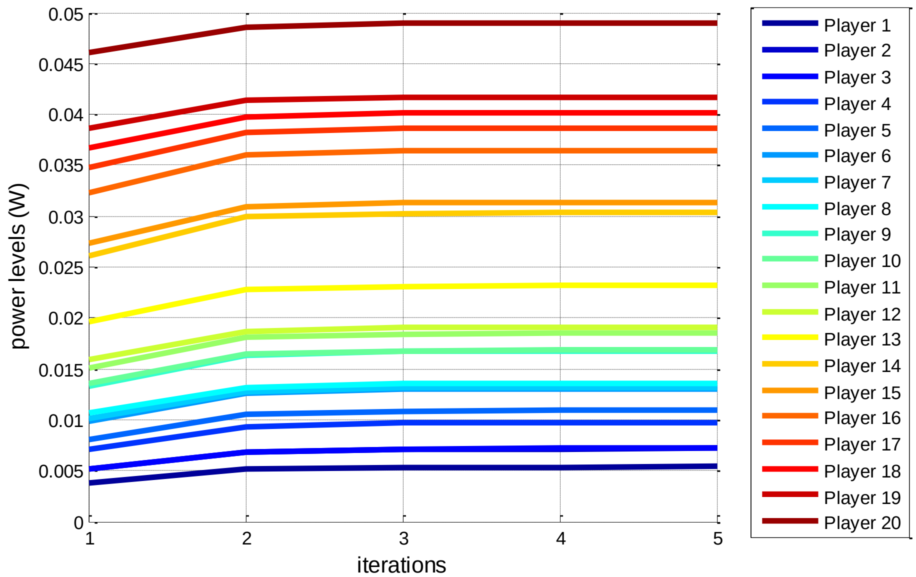


(a) Sequential Play

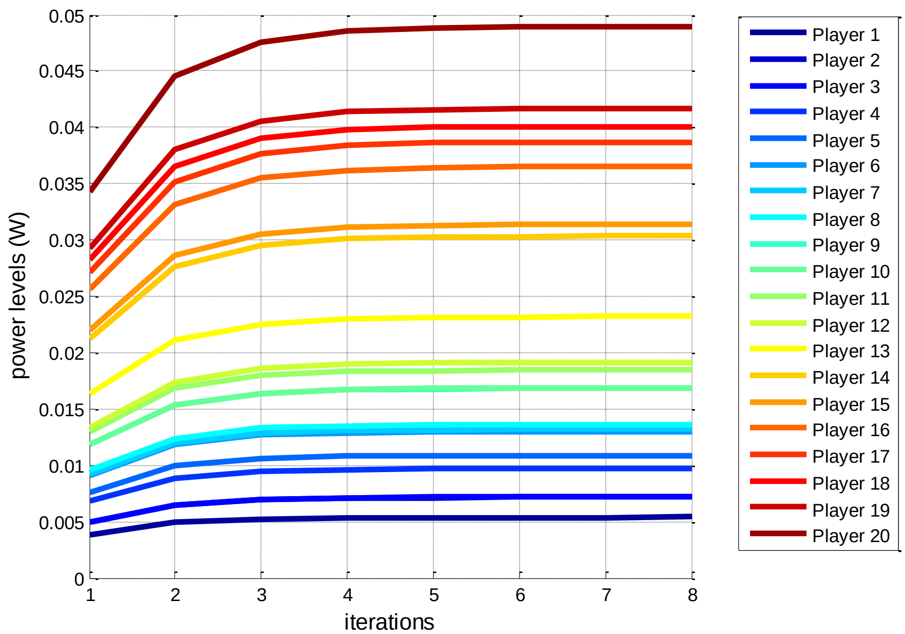


(b) Simultaneous Play

Figure 5.13: Convergence to NE for 10 players



(a) Sequential Play



(b) Simultaneous Play

Figure 5.14: Convergence for 20 players

5.6 No-Regret Game-Theoretic Learning

This section presents the results of implementations of game-theoretic learning. The first subsection shows the performance of a learning player playing against an adversary using the best response to the previous strategy. The results of the learning player playing against a deterministic and probabilistic adversary are then given.

5.6.1 Best Response to the Previous Strategy of the Opponent (BRP)

Figure 5.15 and Figure 5.16 show the strategies of an adversarial opponent together with those of a learning player over 50 iterations. Figure 5.15 illustrates the case where the opponent is deterministic and Figure 5.16 illustrates the case where the opponent is non-deterministic.

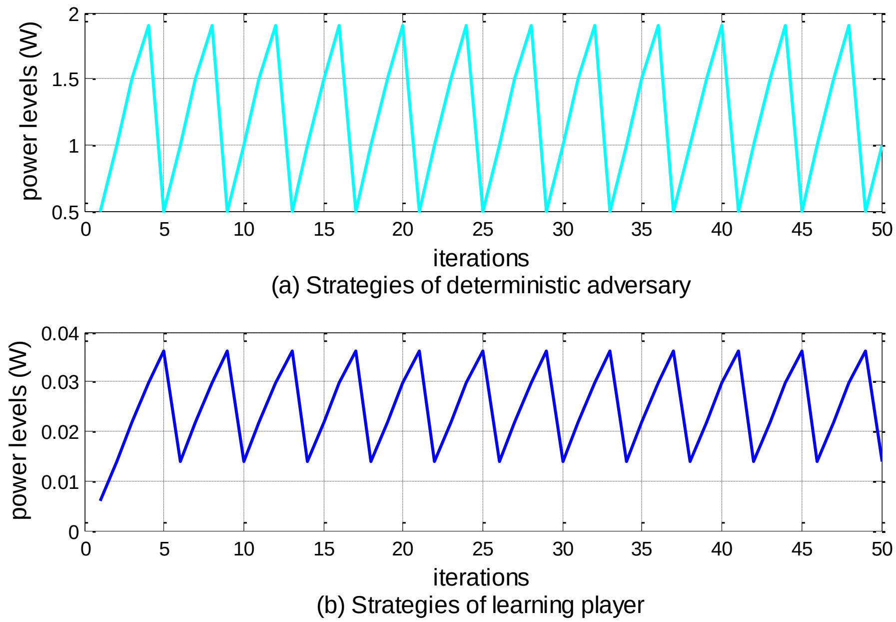


Figure 5.15: Learning players BRP response to deterministic adversary

In both Figure 5.15 and Figure 5.16, the player using the best response dynamic (BRP) is seen to utilize strategies which are best responses to the play of the adversarial opponent in the previous iteration. This is evidenced by the fact that the shape of the graph of the strategies of the learning player follows the shape of the graph of the strategies of the adversarial player, only that they are delayed by one iteration. BRP algorithm does not use any significant history and was the base case used to gauge the performance of the other learning algorithms given that it forms the fundamental implementation of IWF.

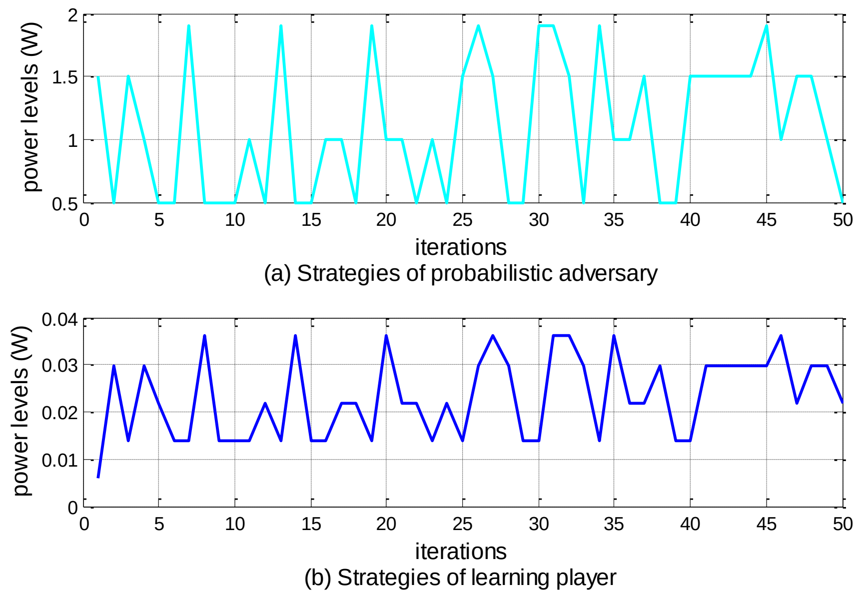


Figure 5.16: Learning player's BRP response to probabilistic adversary

5.6.2 Learning Against a Deterministic Adversary

The learning algorithms were implemented in turn and used against an adversary playing with a deterministic strategy as shown in Figure 5.15(a). Table 5.10 shows the utilities accrued when using the basic IWF as well as the different learning algorithms. Table 5.10 also shows the eventual data rates based on the utilities and the average power. The average utility and average power were the averages taken over 50 iterations that were used for the learning.

Table 5.10: Utilities using the different learning algorithms

	Cumulative Utility ($\times 10^7$ b/J)	**Average Utility ($\times 10^5$ b/J)**	**Average Power (W)**	**Data Rate (b/s)**	**Improvement over BRP (%)**
BRP Player (IWF)	1.0469	2.0939	0.0248	5192.87	N/A
Learning using RMA	1.3338	2.6675	0.0237	6321.98	27.39
Learning using HA	1.3830	2.7660	0.0249	6887.34	32.10
Learning using AWM	1.0469	2.0939	0.0248	5192.87	0.00
Learning using HMA	1.7555	3.5111	0.0240	8426.64	67.68

The player using BRP against the deterministic adversarial player achieved a cumulative utility of 1.0469 x 10^7 b/J over 50 iterations. The player that was able to best learn the strategy of the adversarial player was the player employing the HMA. The HMA yielded a cumulative utility of 1.7555 x 10^7 b/J which was an improvement in the cumulative utility of 0.7086 x 10^7 b/J. This was a 67.68% improvement over the base case. This was the best improvement. On the other hand learning using AWM did not register any improvement over BRP. The percentage improvements were calculated based on the average utilities.

For the HMA, a variation was done on the length of the recent history of stage games (the pattern) being matched with the rest of the previous stage games. This was done in the case where S_2 = {0.5 W, 1 W, 1.5 W}, S_2 = {0.5 W, 1 W, 1.5 W, 1.9W} and S_2 = {0.5 W, 0.9 W, 1.3 W, 1.6 W, 1.9 W}. The results of the variation are illustrated in Table 5.11, Table 5.12 and Table 5.13, respectively.

Table 5.11: Effects of pattern length variation with strategy space size of 3

	pattern length								
	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Cumulative Utility (10^7 b/J)	2.01	1.99	1.96	1.93	1.91	1.88	1.84	1.83	1.80
Average Utility (10^5 b/J)	4.02	3.98	3.92	3.86	3.83	3.75	3.69	3.66	3.60

Table 5.12: Effects of pattern length variation with strategy space size of 4

	pattern length								
	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Cumulative Utility (10^7 b/J)	1.72	1.76	1.73	1.70	1.67	1.64	1.62	1.58	1.55
Average Utility (10^5 b/J)	3.44	3.51	3.47	3.39	3.34	3.28	3.25	3.16	3.11

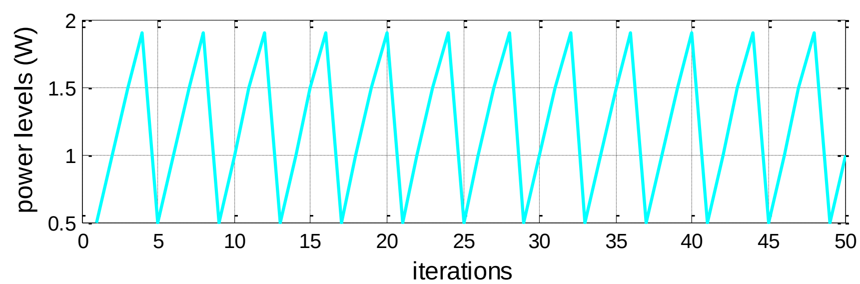
Table 5.13: Effects of pattern length variation with strategy space size of 5

	pattern length								
	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Cumulative Utility (10^7 b/J)	1.64	1.68	1.68	1.66	1.62	1.60	1.57	1.56	1.54
Average Utility (10^5 b/J)	3.29	3.37	3.35	3.32	3.25	3.21	3.14	3.12	3.09

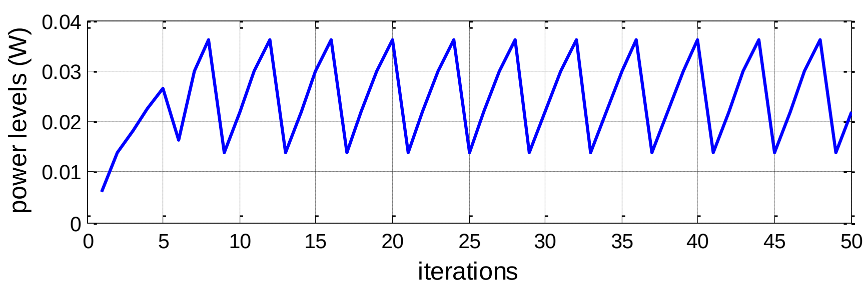
When the strategy space size is 4, the highest utility for the HMA occurred when the pattern length was 3 (this is the utility used for comparison purposes in Table 5.10). The utilities for longer patterns were generally lower. Table 5.11 – Table 5.13 show that a pattern length equal to the strategy space size yields an average utility close to the highest average utility.

The lower cumulative utilities at higher pattern lengths were attributed to the specific implementation of the HMA; based on the implementation of the HMA, the greater the pattern length the more iterations needed before the HMA can be effectively employed i.e. its activation is delayed by a number of iterations proportional to the pattern length. However, it was seen that most of the cumulative utilities of the HMA were still greater than those of the other learning algorithms.

The variation of the strategies for the learning player as well as the deterministic adversarial player are illustrated in Figure 5.17. The pattern length being employed is 4. It is seen that after 6 iterations the strategies of the learning player correspond to those of the adversarial player; the peaks and valleys occur at the same iteration points unlike in the case of best response (Figure 5.15) where there was a delay of one iteration in the learning player to correspond to the adversarial player.



(a) Strategies of the adversarial player



(b) Strategies of the learning player

Figure 5.17: Learning player’s HMA response to deterministic adversary

For the RMA, the history size over which the cumulative regret is calculated was varied. The results are shown in Table 5.14. It was seen that the maximum cumulative utility occurred with a history size of 8. The strategy space for Table 5.14 was S_2 = {0.5 W, 1 W, 1.5 W, 1.9 W}.

Table 5.15 shows the variation when the strategy space was S_2 = {0.5 W, 1 W, 1.5 W}, i.e. when the strategy space size was 3.

Table 5.14: Utility values with varying history size for RMA with strategy space size of 4

	History Size											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**20**	**30**
Cumulative Utility (10^7 b/J)	1.05	0.90	0.92	1.16	1.18	1.03	1.06	1.33	1.25	1.06	1.18	1.11
Average Utility (10^5 b/J)	2.09	1.80	1.84	2.32	2.36	2.06	2.11	2.67	2.50	2.12	2.36	2.23

Table 5.15: Utility values with varying history size for RMA with strategy space size of 3

	History Size											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**20**	**30**
Cumulative Utility (10^7 b/J)	1.15	1.07	1.55	1.35	1.22	1.50	1.37	1.30	1.47	1.40	1.35	1.38
Average Utility (10^5 b/J)	2.30	2.14	3.10	2.69	2.45	2.99	2.74	2.60	2.95	2.80	2.69	2.77

5.6.3 Learning Against a Probabilistic Adversary

Table 5.16 shows the utilities accrued when using the different algorithms when playing against a player who chooses his strategies from the strategy space with certain probabilities. The average utility and average power were the averages taken over 50 iterations that were used for the learning. The strategy space of the adversarial player is:

$$S_2 = \{0.5\text{ W}, 1\text{ W}, 1.5\text{ W}, 1.9\text{ W}\}. \quad (5.4)$$

The strategy chosen in each iteration is $s_i \in S_i$ with each s_i occurring with probability of 0.25.

Table 5.16: Utility values in the probabilistic case

	Cumulative Utility ($X\ 10^7$ b/J)	**Average Utility ($X\ 10^5$ b/J)**	**Average Power (W)**	**Data Rate (b/s)**	**Improvement over BRP (%)**
BRP Player (IWF)	1.2020	2.4039	0.0259	6226.101	N/A
Learning using RMA	1.3372	2.6745	0.0246	6579.27	11.26
Learning using HA	1.3925	2.7850	0.0265	7380.25	15.85
Learning using AWM	1.2508	2.5016	0.0285	7129.56	4.06
Learning using HMA	1.332	2.664	0.0242	6446.88	10.82

In Table 5.16, each of the algorithms were played against the adversarial player 4 times. The utilities accrued each time differed slightly due to the randomization of the strategies chosen by the adversarial player. Therefore, the average values over the four runs were used.

As Table 5.16 shows, the HA algorithm performs best against the probabilistic adversarial player. The HA algorithm achieved a cumulative utility of 1.39×10^7 b/J whereas the BRP player achieved cumulative of 1.202×10^7 b/J. Therefore, the improvement of HA over BRP in the probabilistic case was 15.85%, which was the best improvement. The percentage improvements were calculated based on the average utilities. The players using the RMA, AWM and HMA algorithm also generally achieved utilities higher than the BRP algorithm, which was the base case for the evaluation of the other algorithms. The graphs for the strategies of the probabilistic adversarial player and the player using the HA are shown in Figure 5.18. After a number of iterations the strategies of the learning player using HA converge to a level that depends on the probability distribution of the individual strategies of the adversarial player.

The HMA algorithm performed well in the case of the deterministic adversary. However, in this case of a probabilistic adversary the HMA algorithm was seen not to be the best performer. When the length of the pattern used for history matching in the HMA was varied, it was noticed that there was no improvement in the utility accrued. This is illustrated in Table 5.17. A pattern length of 5 achieved the highest utility. This is the pattern length used for comparison purposes in Table 5.16.

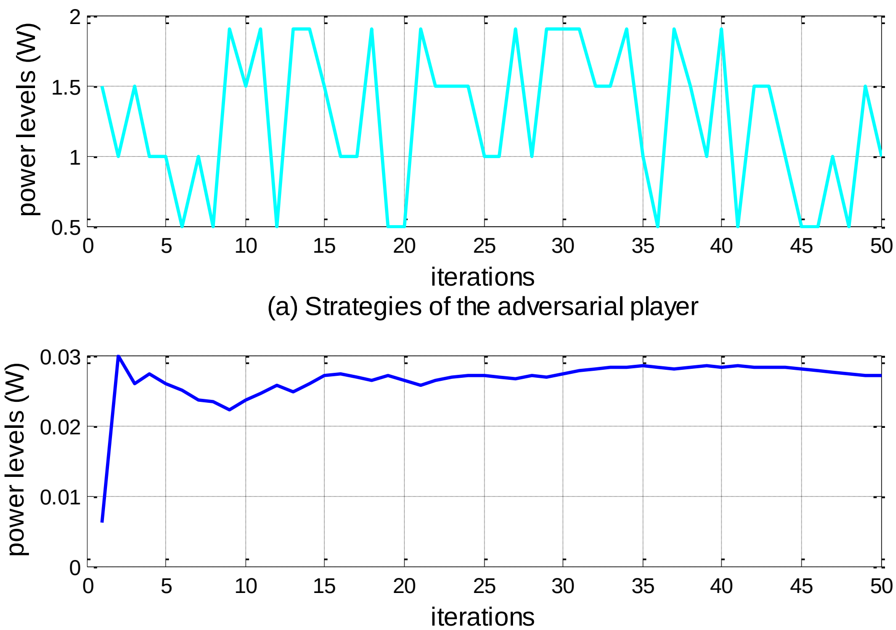


Figure 5.18: Learning player's HA response to probabilistic adversary

Table 5.17: Variation of the pattern length for HMA

	Pattern length								
	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Cumulative Utility (10^7 b/J)	1.25	1.10	1.26	1.33	1.20	1.15	1.17	1.20	1.11
Average Utility (10^5 b/J)	2.51	2.20	2.52	2.66	2.39	2.31	2.35	2.41	2.22

5.7 Comparison of BRP algorithm and No-Regret Learning in Converging to NE

The best response dynamic that has been hitherto employed to investigate the NE characteristics was compared to the implementation of RMA learning algorithm. The RMA was chosen as the basis for comparison in this case as it appeared to be the most basic of the learning algorithms. The convergence characteristics of the learning algorithm in the case of 10 players are depicted in Figure 5.19.

Comparing Figure 5.19 to Figure 5.1, it is seen that for 10 players, using the RMA learning algorithm, the play converges to the same NE as in the case of the best response dynamic (Figure 5.1). However, convergence to NE takes place after 17 iterations as compared to the case of the best response dynamic in sequential and simultaneous play, which takes 3 and 4 iterations, respectively. This means that IWF (through BRP) converges more than five times faster than learning in sequential play.

Therefore, it is seen that the learning algorithms converge to pure strategy NE in games for which the pure strategy NE exists. However, the learning algorithm converges comparatively slowly.

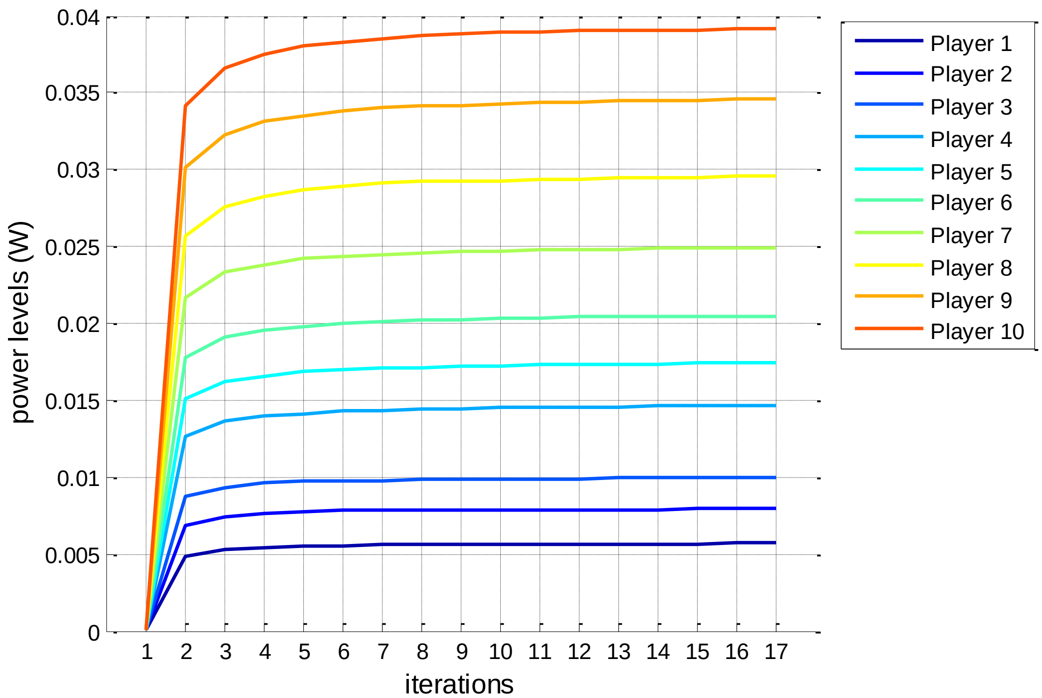


Figure 5.19: Convergence to NE of 10 players all using a learning algorithm

5.8 General Comments on BRP, HMA and HA

It has been seen that the learning algorithms are capable of adapting well to the strategies of a player who does not play according to the best response dynamic. By extension, the learning algorithms are also capable of adapting to changes in the network environment. The learning algorithms adapt better than the iterative water-filling, exemplified by the BRP algorithm, which was the basic algorithm used to arrive at and maintain the NE.

Among the learning algorithms, the HMA was seen to perform better in the case where the adversarial player changes strategies in a deterministic fashion. In the case where the adversarial player changes strategies in a probabilistic fashion, the HA was seen to perform better than the other algorithms.

The BRP algorithm was seen to have the advantage of faster convergence to NE in the scenario where the players are all playing their best responses to the strategies of the other players. In such cases the learning algorithms take many more iterations to converge to the same equilibrium point.

Therefore, whereas BRP converges faster, after equilibrium the learning algorithms adapt to changes better.

5.9 Performance of the Proposed Hybrid-Adaptive Algorithm

Figure 5.20 shows the strategies of the modes of play adopted by the adversarial player over 150 stage games as well as the corresponding responses of the learning player. In this case 150 was selected for the number of stage games to merely present more iterations in which the learning algorithms could be studied and tested.

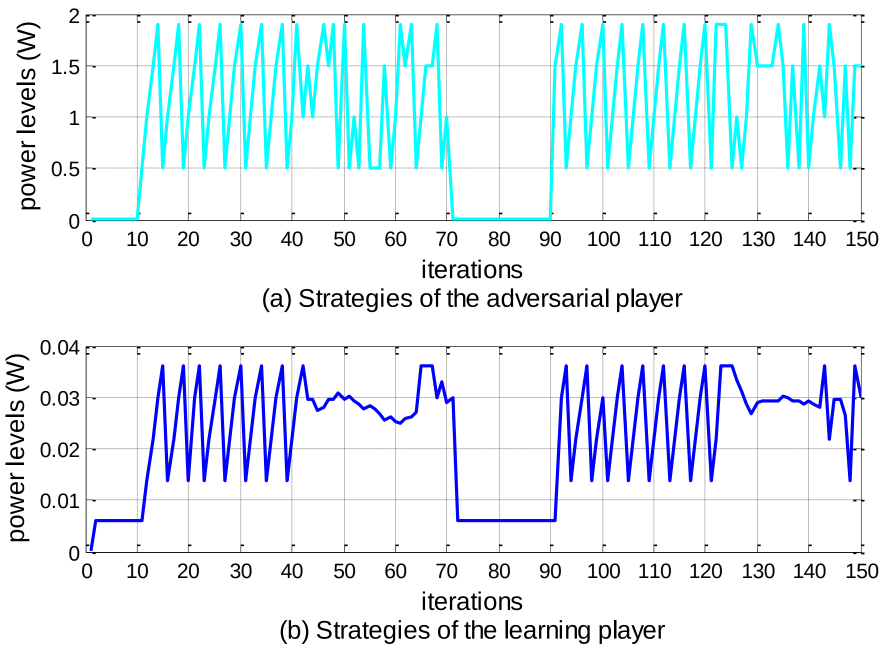


Figure 5.20: Strategies of the varying adversary and the adapting learning player

For purposes of clarity Figure 5.21 shows the strategies of the adversarial player over the first 10 stage games. It is noted that after the first three iterations there is a convergence to NE. This is due to the fact that the learning player is also playing according to BRP, which typically has fast convergence. There is also a convergence to the same NE from iteration 71 – 90 since the adversarial player is using BRP which prompts the same response from the learner.

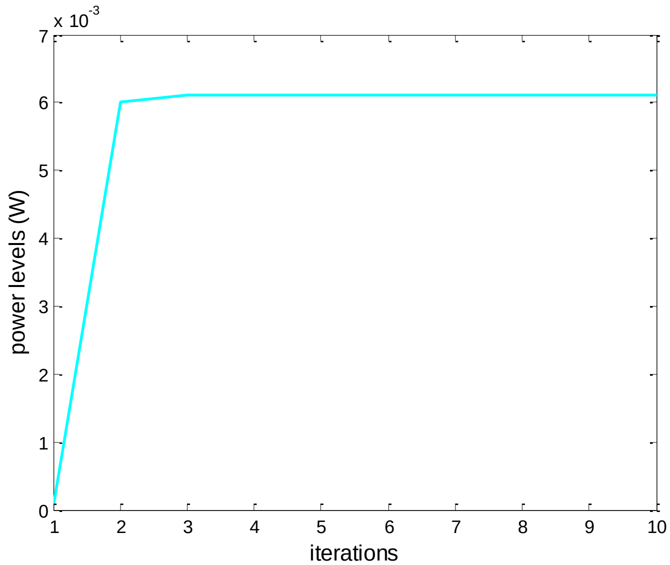


Figure 5.21: Convergence to NE in the first 10 iterations

As shown in Figure 5.20(a), the adversarial player employs different modes of play. Initially it plays according to BRP for 10 stage games. It then changes to a deterministic mode of play for 30 stage games followed by a probabilistic mode of play for another 30 stage games. This sequence of BRP, deterministic and probabilistic modes of play is then repeated for the next 80 stage games: BRP (iteration 71 – 90), deterministic mode (iteration 91 – 120) and probabilistic mode (iteration 121 – 150). Figure 5.20(b) shows the strategies that the learning player adopts based on the detected modes of play of the adversarial player. It is seen that in the first 10 stage games the learning player also adopts a BRP mode of play. In this case, the play converges to a NE, with each player having a transmit power of 0.0061 W.

The next stage games (11 – 40) show the learning player's response to the adversarial player employing a deterministic mode of play. It is seen that once the learning player detects this strategy the HMA algorithm enables the learning player to follow closely the periodic variations of the adversarial player, thus maximizing utility. It is noticed that between iteration 11 and 20, the peaks and valleys of the learning player trail those of the adversarial player and from iteration 20 onwards they coincide. This is due to the fact that the learning player takes 9 iterations to acquire sufficient history of the deterministic play and only then is it able to make more accurate predictions of the adversarial player's strategies.

In stage games 41 – 70, the learning player's response to the adversarial player employing a probabilistic strategy is shown. Within a few iterations the learning player detects the change in the mode of play from deterministic to a probabilistic play and correspondingly adapts its mode of play to the HA. It is noticed that between iteration 140 and 150 the learning player detects the mode of play of the adversary wrongly. This is manifested in the glitches in which the learning player momentarily breaks away from HA even though the adversarial player is still using a probabilistic mode of play. This is an error in the detection process of the hybrid-adaptive learning algorithm and offers a potential area of improvement. Iterations 71 to 150 also show the adversarial player changing its mode of play from BRP to a deterministic mode and finally to a probabilistic mode. The learning player selects the best algorithm to employ given the detected mode of play of the adversarial player.

5.9.1 Comparison of Individual Learning Algorithms and the Proposed Algorithm

The individual learning algorithms were also used against the adversarial player employing the different modes of play for comparison purposes. Table 5.18 shows the cumulative and average utilities in the cases of a learning player using the proposed algorithm (hybrid-adaptive algorithm) as well as the learning player separately using the BRP algorithm, HMA algorithm and HA.

Table 5.18: Utilities for different responses to a varying adversary over 150 iterations

	Cumulative Utility ($\times 10^7$ b/J)	**Average Utility ($\times 10^5$ b/J)**	**Average Power (W)**	**Data Rate (b/s)**	**Proposed Algorithm's Improvement (%)**
BRP Algorithm (IWF)	6.6220	4.4147	0.0214	9447.458	12.65
HMA	6.8156	4.5437	0.0211	9587.207	9.45
HA	4.4531	2.9688	0.0118	3503.184	67.52
Proposed Algorithm	7.4598	4.9732	0.0228	11338.896	N/A

From Table 5.18 it is seen that the adaptive algorithm performs better than BRP, HMA and HA algorithms taken separately. Using the adaptive algorithm, there is an improvement in the average utility of 12.65% over BRP, 9.45% over HMA and 67.52% over HA.

Figure 5.22 shows the variations of the modes of play of the adversarial player and the response of a learning player using only the BRP algorithm. The learning player essentially has a best response to the previous play of the adversarial player and this is seen in the general shape of the plot of the strategies: the shape of the learner's graph is similar to the adversarial opponent's graph and it mainly differs in the fact that it is delayed by one iteration and the magnitudes are much smaller. In the stages where the adversary is also employing BRP (iterations 1 – 10 and 70 – 90) the play converges quickly to the NE. When the adversary adopts other strategies, there is a departure from the NE. The average utility achieved when using BRP alone against the changing adversarial opponent was 4.4147 x 10^5 b/J.

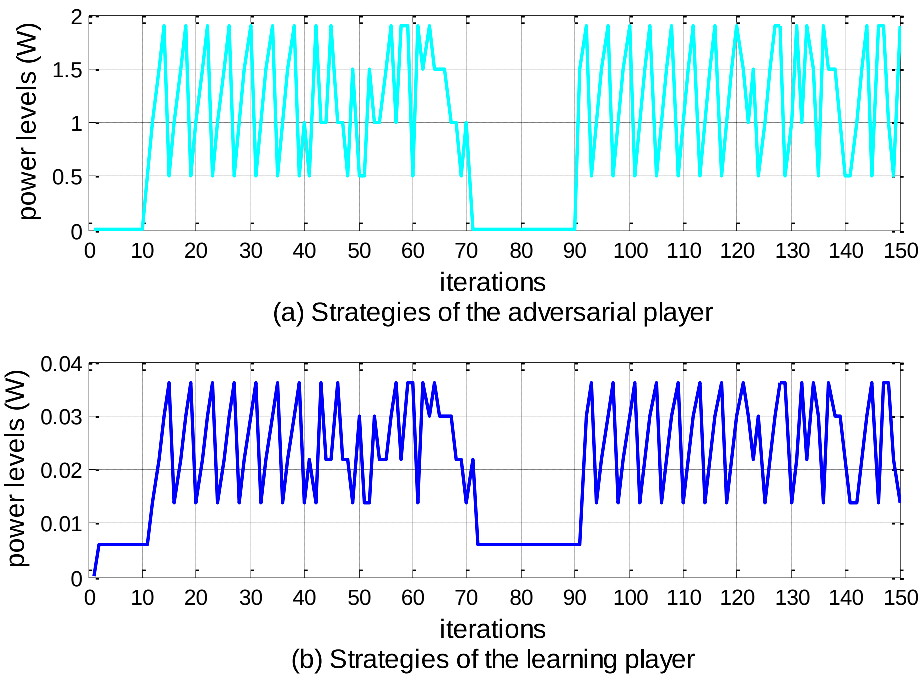
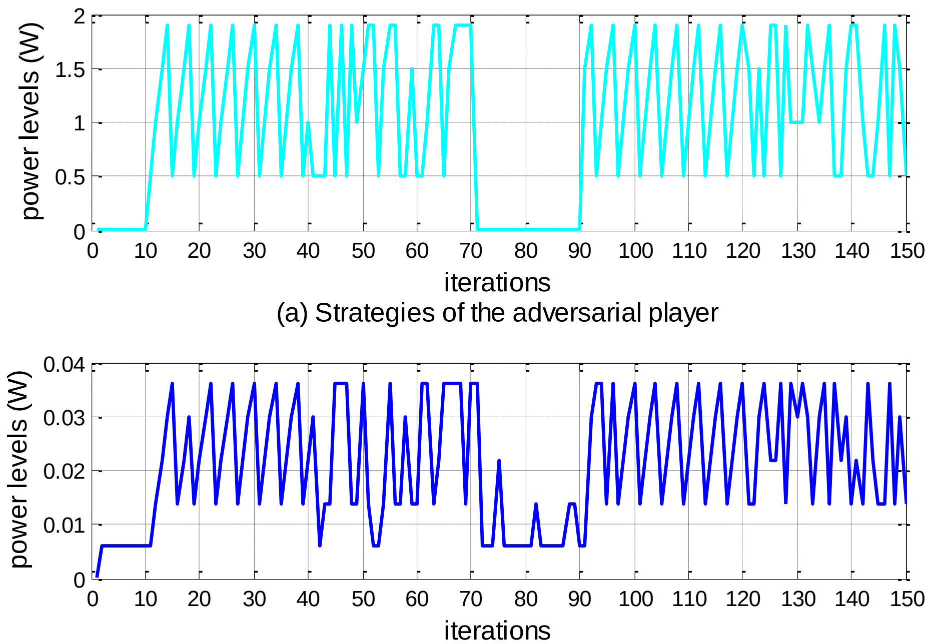


Figure 5.22: Strategies of the varying adversary and the learning player using BRP alone

Figure 5.23 shows the corresponding responses of learning players using only the HMA. In this case the general shape of the graph of the learning player is not similar to the shape of the graph of the adversarial player. The HMA matches the recent strategies to the history of strategies available in order to predict the next strategy of the adversary. In this case the play never arrives at the NE. This suggests that the HMA is not suitable in a situation where convergence to the NE is desired. The HMA used alone achieves an average utility of 4.5437 x 10^5 b/J, which is higher when compared to that of BRP alone.



(a) Strategies of the adversarial player

(b) Strategies of the learning player

Figure 5.23: Strategies of varying adversary and learning player using HMA alone

Figure 5.24 show the corresponding responses of learning players only using the HA. In this case the shape of the graph for the learner's strategies does not follow the shape of the adversarial player's graph. The play achieves the NE when the adversarial player is using BRP (iterations 1 – 10 and 70 – 90). The learner reacts significantly mainly when the adversarial player is using a probabilistic strategy (iterations 41 – 70 and 120 – 150). The player using HA only achieved an average utility of 2.9688 x 10^5 b/J which was lower than in the case of HMA alone and BRP alone.

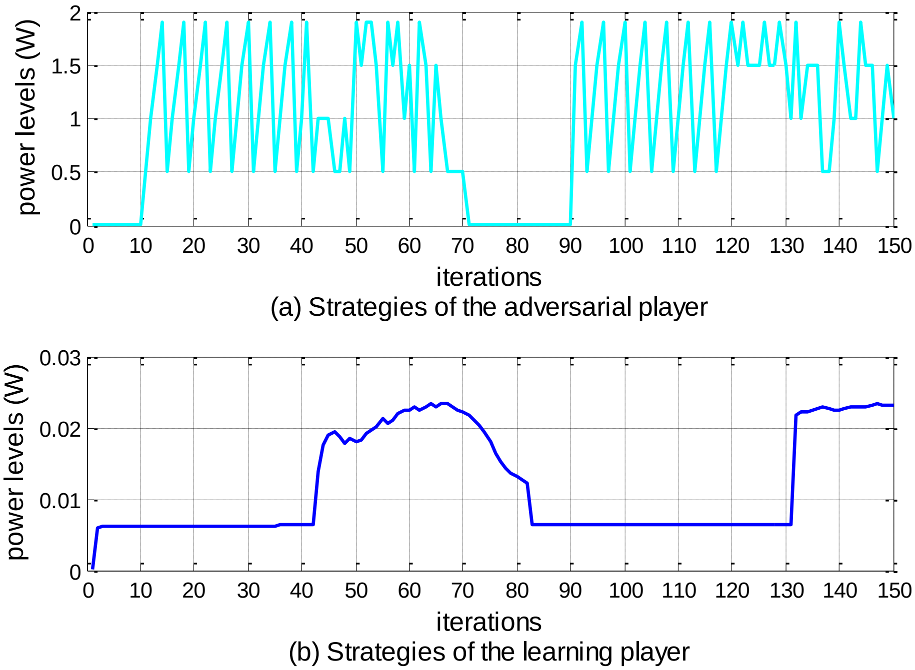


Figure 5.24: Strategies of varying adversary and the learning player using HA alone

5.9.2 Improvements Resulting from the Hybrid-Adaptive Algorithm

The hybrid-adaptive algorithm brings out some improvements on related works. Nie and Comaniciu [16] treat convergence to the Nash Equilibrium (based on a potential function) and the game-theoretic learning in isolation. Several other authors also treat iterative water-filling and learning in isolation [6][34][18][35][19]. The algorithm presented here is an interface of the two, which offers improvements. Ifeh's [30] simulation results also show a treatment of the techniques in isolation. His main findings are a comparison of water-filling and no-regret learning giving the strengths and weaknesses of these techniques taken separately. Although he proposes a hybrid scheme, the implementation and simulation data is left for further research. The novel hybrid-adaptive algorithm in this research interfaces iterative water-filling and learning with simulation data; furthermore, learning is harnessed in a dynamic manner, meaning that the algorithm has the ability to detect and adapt to the mode of play of an adversarial player (external environment). This is achieved by an analysis of several learning algorithms and the corresponding incorporation of several learning algorithms into the hybrid-adaptive algorithm.

5.9.3 Distributed Transmit-Power Control through the Hybrid-Adaptive Algorithm

The hybrid-adaptive algorithm developed is able to perform well when compared to the other algorithms taken separately. The algorithm maximizes the utility accrued by appropriately controlling the transmit-power based on the analysis of the network environment, embodied by the actions of the adversarial player.

When the adversarial payer utilizes BRP, the player using the hybrid-adaptive algorithm also used BRP and the result is that the play ends up in a Nash Equilibrium. This equilibrium is made possible through the repeated control of the transmit-power. If the adversary is not using BRP, the play need not necessarily end up in the Nash Equilibrium but through transmit-power adjustments the learning player mitigates exploitation and optimizes it utility.

6 CONCLUSION AND FURTHER WORK

In this research transmit-power control in cognitive radio networks has been studied and represented as a non-cooperative game-theoretic problem. To achieve the first objective of the research, a cognitive radio network was modeled as a single-cell W-CDMA network with a number of mobile stations representing the players in the game-theoretic framework. The model was simulated using MATLAB. The scenario was a repeated game of infinite horizon. In the cognitive radio network the players all attempted to maximize their utilities in a distributed manner by appropriately adjusting their powers, which formed the strategy space. The utility of each player based on the utility function employed represented the bits transmitted per joule of energy. A crucial aspect of the game-theoretic framework was seen to be the concept of the Nash Equilibrium (NE), which represented a stable operating point.

In order to arrive at the Nash Equilibrium the iterative water-filling (IWF) algorithm was implemented by employing the best response dynamic (BRP). This helped achieve the second objective of the research. Various characteristics of the NE were investigated such as the convergence speed, the power levels, the SINR and the utility at NE. Two modulation schemes were also investigated with respect to the NE and its characteristics. The modulation schemes employed were QPSK and FSK. It was found that when using QPSK, the convergence to NE was much faster than in the case of FSK. For the same positions of mobile stations (players) it was also seen that those using QPSK achieved a higher utility than those using FSK. The developed techniques were modeled on a mobile network employing CDMA (QPSK) but they can be employed in other kinds of networks, which may use QPSK, FSK or other modulation schemes.

The Pareto efficiency of the Nash Equilibrium arrived at was also assessed. It was seen that the equilibrium does not represent a Pareto-optimal power vector. An algorithm for Pareto-Improvement was developed and implemented. The algorithm helped achieve a higher overall utility as compared to the Nash Equilibrium for the entire system while guaranteeing that the utility for all players was at least equal to their utilities at the Nash Equilibrium. Based on the results of the algorithm an equation useful for directly finding the Pareto-superior power vector was then developed. This method was seen to offer improvements to other methods used for finding Pareto-superior power vectors to the Nash Equilibrium.

In carrying out the game-theoretic analysis in the cognitive radio network, sequential play and simultaneous play were also compared. It was seen that the sequential play converged faster than simultaneous play given that in sequential play the users had knowledge of the opponents' moves in the current stage games.

Since IWF does not prevent against possible exploitation of players by their opponents, game-theoretic learning was also implemented in an attempt to mitigate the possible exploitation. Therefore, the third objective of the research was achieved. A number of learning algorithms were implemented and these included a regret-matching algorithm (RMA), the hedging algorithm (HA), weighted majority algorithm (WMA) and the historic matching algorithm (HMA). The learning algorithms resulted in a convergence to the same NE as in the case of IWF. The learning algorithms were also able to adapt better to changes in the overall network environment. These changes were represented by an adversarial player which had different modes of playing: in some instances the adversary played using BRP, in some instances using a deterministic mode of play and in some instances using a probabilistic mode of play. The learning algorithms were found to generally adapt better than IWF (implemented using BRP) to the changes is the mode of play of the adversarial player. This better adaptation was manifested by higher utilities.

A deeper investigation was made as to the best mode of play for the player employing a learning algorithm (the learning player) when faced with the different modes of player of the adversarial player. It was found that when the adversary employed BRP, then BRP was also the appropriate response for the learning player; when the adversary employed a deterministic strategy, then using HMA was the appropriate response by the learning player; when the adversary employed a probabilistic strategy, then HA was the appropriate response by the learning player.

The research led to the development of a hybrid and adaptive transmit-power control algorithm. This achieved the fourth objective of the research. The hybrid nature of the overall algorithm was in the interfacing of the IWF and game-theoretic learning. The hybrid algorithm drew from the strengths of both algorithms. In the hybrid, IWF was run first followed by the learning component, given that IWF has faster convergence. The adaptive nature of the overall algorithm was in the fact that it changed among different learning algorithms depending on the perceived behaviour of other players in the network. This novel algorithm offered an improvement on the previous treatments of iterative water-filling and game-theoretic learning.

6.1 Main Contributions

The following is a summary of the main contributions resulting from the research:

i. A characterization and comparison of the Nash Equilibrium was given with respect to:
   a. Two modulation schemes: QPSK and FSK.
   b. Sequential and simultaneous play.
   c. Game-theoretic framework with learning and without learning.

ii. An algorithm to find a Pareto-superior power vector to the Nash Equilibrium was developed and implemented. This was further improved on by developing an equation by which the Pareto-superior strategy could be directly arrived at.

iii. The appropriate responses of a learning player (in terms of the learning algorithm to employ) to different modes of operation of other players were determined.

iv. A novel hybrid-adaptive algorithm was developed and implemented. The hybrid was an interface of iterative water-filling and game-theoretic learning. The algorithm was able to adapt between the Hedging Algorithm and Historic Matching Algorithm based on the behaviour of the adversary. The hybrid-adaptive algorithm was able to achieve fast convergence as well as mitigate exploitation. It achieved an improvement in the average utility of 12.65% over BRP, 9.45% over HMA and 67.52% over HA.

6.2 Further Work

Part of the process of selecting among learning algorithms is a detection of the mode of play of the adversarial player representing the general network environment. A deeper investigation needs to be made into this process of detection so as to make it improve on its accuracy. Accurate detection of the behaviour of other players results in the most appropriate learning algorithm used as a response of the learning player and leads to a higher overall utility.

The Pareto-superior strategy arrived at in this research was not an equilibrium and in an environment of non-cooperation, the situation can revert to the Nash Equilibrium, which is a more stable operating point. Further work will consist in an implementation of punishment in repeated games, which can act as a motivating factor for all players to maintain the Pareto-superior power vector.

REFERENCES

[1] Federal Communications Commission, "Report of the Spectrum Efficiency Working Group," Nov. 2002.

[2] Deloitte & Touche, "Competition Study - The Broadcasting Industry in Kenya," Mar. 2012.

[3] S. Haykin, "Cognitive Radio: Brain-Empowered Wireless Communications," *IEEE Journal on Selected Areas in Communications.*, vol. 23, no. 2, pp. 201–220, Feb. 2005.

[4] J. Mitola and G. Q. Maguire, "Cognitive Radio: Making Software radios more Personal," *IEEE PCS Magazine*, vol. 6, no. 4, pp. 13–18, Aug. 1999.

[5] I. F. Akyildiz, W. Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt Generation/Dynamic Spectrum Access/Cognitive Radio Wireless Networks: a Survey," *Elsevier Computer Networks Journal*, vol. 50, no. 13, pp. 2127–2159, May 2006.

[6] B. H. Ellingsæter, "Cognitive Radio: Interference Management and Resource Allocation," MSc Thesis, Norwegian University of Science and Technology, Norway, 2010.

[7] L. Lu, X. Zhou, U. Onunkwo, and G. Y. Li, "Ten Years of Research in Spectrum Sensing and Sharing in Cognitive Radio," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 28, 2012.

[8] J. Palicot, J. Mitola, Z. Lei, and F. K. Jondral, "Special Issue on 10 Years of Cognitive Radio: State-of-the-art and Perspectives," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 214, 2012.

[9] IEEE Standards Association, "Part 22: Cognitive Wireless RAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Policies and Procedures for Operation in the TV Bands." 2011.

[10] IEEE Standards Association, "IEEE 802.22 WRAN WG Website," 2011. [Online]. Available: http://www.ieee802.org/22/. [Accessed: 15-Nov-2013].

[11] Federal Communications Commission, "FCC 03-289," Nov. 2003.

[12] Y. Zhang, J. Zheng, and H. Chen, *Cognitive Radio Networks*. CRC Press Taylor and Francis Group, 2010.

[13] Q. Zhao and B. M. Sadler, "A Survey of Dynamic Spectrum Access," *IEEE Signal Process. Magazine*, vol. 79, pp. 79–89, 2007.

[14] C. Peng, H. Zheng, and B. Y. Zhao, "Utilization and Fairness in Spectrum Assignment for Opportunistic Spectrum Access," *ACM Mobile Networks and Applications MONET*, vol. 11, no. 4, pp. 555–576, 2006.

[15] H. Zheng and C. Peng, "Collaboration and Fairness in Opportunistic Spectrum Access," in *IEEE International Conference on Communication Technology*, vol. 5, pp. 3132–3136, 2005.

[16] N. Nie and C. Comaniciu, "Adaptive Channel Allocation Spectrum Etiquette for Cognitive Radio Networks," in *IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, Baltimore, pp. 269–278, 2005.

[17] R. Etkin, A. Parekh, and D. Tse, "Spectrum Sharing for Unlicensed Bands," *IEEE Journal on Selected Areas in Communications.*, vol. 25, no. 3, pp. 517–528, 2007.

[18] D. D. Yu and J. M. Cioffi, "Iterative Water-filling for Optimal Resource Allocation in OFDM Multiple-Access and Broadcast Channels," in *IEEE GLOBECOM*, San Francisco, pp. 1–5, 2006.

[19] W. Yu, G. Ginis, and J. M. Cioffi, "Distributed Multiuser Power Control for Digital Subscriber Lines," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 5, pp. 1105–1115, 2002.

[20] W. Yu, "Competition and Cooperation in Multi-User Communication Environments," Ph. D. dissertation, Stanford University, California, 2002.

[21] R. Menon, R. M. Buehrer, and J. H. Reed, "Outage Probability Based Comparison of Underlay and Overlay Spectrum Sharing Techniques," in *IEEE DySPAN*, Baltimore, pp. 101–109, 2005.

[22] E. Z. Tragos, S. Zeadally, A. G. Fragkiadakis, and V. A. Siris, "Spectrum Assignment in Cognitive Radio Networks: A Comprehensive Survey," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1108–1135, 2013.

[23] Federal Communications Commission, "FCC 07-78A1," 2007.

[24] C. D. Imlay, "Comments of ARRL, The National Association for Amateur Radio (ET Docket No. 03-237)." American Radio Relay League, 2004.

[25] Committee on Communication Policy, "Clarifying Harmful Interference will Facilitate Wireless Innovation." IEEE-USA, 2012.

[26] B. Zayen, M. Haddad, A. Hayar, and G. E. Øien, "Binary Power Allocation for Cognitive Radio Networks with Centralized and Distributed User Selection Strategies," *Physical Communication.*, vol. 1, no. 3, pp. 183–193, 2008.

[27] A. B. MacKenzie and L. A. DaSilva, *Game Theory for Wireless Engineers*. Morgan and Claypool, 2006.

[28] Y. Freund and R. E. Shapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *Journal of Computer and System Sciences*, vol. 55, pp. 119–139, 1997.

[29] Q. Qi and Y. Yang, "An Efficient Water-Filling Algorithm for Power Allocation in OFDM-Based Cognitive Radio Systems," in *IEEE International Conference on Systems and Informatics*, pp. 2069–2073, 2012.

[30] K. Ifeh, "Transmit-Power Control for Cognitive Networks: Challenges, Requirements and Options," M. Sc. Thesis, University of Witwatersrand, 2012.

[31] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*, MIT Press, 1994.

[32] M. Felegyhazi and J. P. Hubaux, "Game Theory in Wireless Networks: A Tutorial," EPFL, Switzerland, Technical Report LCA-REPORT-2006-002, 2006.

[33] J. D. Neel, "Analysis and Design of Cognitive Radio Networks and Distributed Radio Resource Management Algorithms," Ph.D. Dissertation, Virginia Polytechnic Institute and State University, Virginia, 2006.

[34] C. U. Saraydar, N. B. Mandayam, and D. J. Goodman, "Efficient Power Control via Pricing in Wireless Data Networks," in *IEEE Wireless Communications and Networking Conference*, pp. 291–303, 2000.

[35] K. W. Shum, K. K. Leung, and C. W. Sung, "Convergence of Iterative Waterfilling Algorithm for Gaussian Interference Channels," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 6, pp. 1091–1100, 2007.

[36] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, Inc., 1991.

[37] D. Fudenberg and D. K. Levine, *The Theory of Learning in Games*. Cambridge, Massachusetts: MIT Press, 1998.

[38] D. Monderer and L. S. Shapely, "Games and Economic Behaviour," *Potential Games*, no. 14, pp. 124–143, 1996.

[39] A. Greenwald and A. Jafari, "A General Class of No-Regret Learning Algorithms and Game-Theoretic Equilibria," in *Computational Learning Theory Conference*, pp. 1–11, 2003.

[40] A. Jafari, A. Greenwald, D. Gondek, and G. Ercal, "On No-Regret Learning, Fictitious Play and Nash Equilibrium," in *International Conference on Machine Learning*, pp. 226–233, 2001.

[41] S. Hart and A. Mas-Colell, "A Simple Adaptive Procedure Leading to Correlated Equilibria," *Econometrica*, vol. 68, no. 5, pp. 1127–1150, 2000.

[42] N. Littlestone and M. K. Warmuth, "The Weighted Majority Algorithm," *Information and Computation*, no. 108, pp. 212–261, 1994.

[43] K. Farkas, T. Hossmann, F. Legendre, B. Plattner, and S. K. Das, "Link quality prediction in mesh networks," *Computer Communications.*, vol. 31, no. 8, pp. 1497–1512, 2008.

[44] A. B. Mackenzie and S. B. Wicker, "Game Theory in Communications: Motivation, Explanation, and Application to Power Control," in *IEEE GLOBECOM*, vol. 2, pp. 821–826, 2001.

[45] D. Goodman and N. Mandayam, "Power Control for Wireless Data," *IEEE International Workshop on Mobile Multimedia Communications.*, pp. 55–63, 1999.

[46] D. Famolari, N. Mandayam, D. Goodman, and V. Shah, "A New Framework for Power Control in Wireless Data Networks: Games, Utility and Pricing." Rutgers University, 2001.

[47] J. G. Proakis, *Digital Communications*, 4th Edition. New York: McGraw-Hill, 2001.

[48] S. Joshi, "Outdoor Propagation Models," *International Journal on Computer Science and Engineering*, vol. 4, no. 2, 2012.

APPENDIX A

Expected Value of SINR at the Nash Equilibrium

The utility function used in the research is given in equation 4.2 and can be written as

$$u_i(p_i, \boldsymbol{p}_{-i}) = \frac{LR}{Mp_i} f(\gamma_i) \tag{A1.1a}$$

where

$$f(\gamma_i) = (1 - 2\ BER)^M \tag{A1.1b}$$

At the Nash Equilibrium there is a mutual best response correspondence, meaning that all users have maximized their utility functions in response to the other users' adjustments. For each player *i* to maximize its utility, it is required that

$$\frac{\partial u_i(p_i, \boldsymbol{p}_{-i})}{\partial p_i} = 0 \tag{A1.2}$$

which gives the maxima of the utility function as well as the condition for the occurrence of the Nash Equilibrium.

$$\frac{\partial u_i(p_i, \boldsymbol{p}_{-i})}{\partial p_i} = \frac{LR}{M}\left(\frac{1}{p_i}\frac{\partial f(\gamma_i)}{\partial p_i} - \frac{f(\gamma_i)}{{p_i}^2}\right) \tag{A1.3}$$

where γ_i is given by equation 4.4. Equation A1.3 yields

$$\frac{\partial u_i(p_i, \boldsymbol{p}_{-i})}{\partial p_i} = \frac{LR}{M}\left(\frac{1}{{p_i}^2}\frac{df(\gamma_i)\gamma_i}{d\gamma_i} - \frac{f(\gamma_i)}{{p_i}^2}\right)$$

$$= \frac{LR}{M{p_i}^2}\left(\frac{df(\gamma_i)\gamma_i}{d\gamma_i} - f(\gamma_i)\right) \tag{A1.4}$$

Therefore, for equation A1.2 to hold, it is required that

$$\frac{LR}{M{p_i}^2}\left(\frac{df(\gamma_i)\gamma_i}{d\gamma_i} - f(\gamma_i)\right) = 0 \tag{A1.5}$$

A solution is given by the values of γ_i that makes $\frac{df(\gamma_i)\gamma_i}{d\gamma_i} - f(\gamma_i) = 0$

The condition for the Nash Equilibrium is therefore now given as

$$\frac{df(\gamma_i)\gamma_i}{d\gamma_i} - f(\gamma_i) = 0 \tag{A1.6}$$

For QPSK, the BER is given by equation 4.3. Substituting the first term of equation 4.3 in equation A1.1b and using equation A1.1b in equation A1.6 yields

$$\gamma_i M \frac{e^{-\gamma_i}}{\sqrt{\pi\gamma_i}}\left(erf\left(\sqrt{\gamma_i}\right)\right)^{M-1} - \left(erf\left(\sqrt{\gamma_i}\right)\right)^{M} = 0 \tag{A1.7}$$

NB: only the first term of equation 4.3 is used because the second term is negligible compared to the first term.

Based on the value of M given in section 4.1.2 and using the numerical technique shown in section B.5 of Appendix B equation A1.7 can be solved for γ_i; this gives the value of γ_i that satisfies the condition for Nash Equilibrium (equation A1.6). This value of γ_i was found to be 5.27 for QPSK.

Value of Thermal Noise Power (σ^2) used in Calculating SINR

The noise spectral density is given by

$$\frac{P}{B} = k_B T \ \ (W/Hz)$$

where P = noise power in Watts

B = Bandwidth

k_B = Boltzmann constant = $1.381 \times 10^{-23} J/K$

T_o = Room temperature in degrees Kelvin = 290 K

Therefore,

$$\frac{P}{B} = 4 \times 10^{-21} \, (W/Hz)$$

In the case of B = 5 MHz,

$$P = \sigma^2 = 4 \times 10^{-21} \times 5 \times 10^{6} = 2 \times 10^{-14} \, W$$

APPENDIX B

Sample Code of the MATLAB Implementations

B1 Game-Theoretic Power Control

```
% Developed by Oscar Ondeng, University of Nairobi. 22-09-15.
% implementation of game-theoretic power control
% WCDMA

% -Best response of each player to other players' actions
% => iterations converging to a NASH EQUILIBRIUM

% ALL PLAYERS WITH THE SAME UTILTIY FUNCTIONS

% -Strategy Space: each player can transmit at power of between 0 and 2W
%  in steps of 0.1

clc;
close all;
clear all;

% initalize game parameters
N = 10;  % number of players
iterations = 300; %  maximum number of iterations during convergence

% distances from the base station (randomly generated)
a = 1; % minimum distance from the base station
b = 2; % maximum distance from the base station
d = sort(a + (b-a).*rand(1, N)); % distances (km)

% Stratety Space
p_max = 2; % maximum power transmission level
step = 0.001; % the steps in which the power levels can increase
p_min = step; % minimum power transmission level
p = ones(1, N) * p_min; % initial power levels

powerTracer = zeros(N,iterations); % to keep track of how long it takes to
converge
convergenceCount = 0; % to count how many iterations are needed for
convergence
ceilingHit = false; % checks whether any player reached p_max

utilityVsPowerTracer = zeros (N, (p_max - p_min) / step + 1);
utilityVsIterationsTracer = zeros(N,iterations);

% loop to iterate until NE convergence
for convergence = 1:iterations

    % initialize vectors
```

```
    br_power = nan(1, N); % power at the best response
    br_utility = nan(1, N); % utility at the best response
    br_snr = nan(1, N); % snr at the best response

    % loop to search for best response for each player
    for i =  1:N %N:-1:1 %

        k = 1; % counter for tracking power trajectory
        ind = 1; % coutner for tracking utility

        % provide initial best response
        if isnan(br_power(i))
            [br_power(i), br_utility(i), br_snr(i)] = utility(i,p,d); %best
response utility - QPSK
%              [br_power(i), br_utility(i), br_snr(i)] = u(i,p,d*1000); %
best response utility - FSK
        end

        % search through strategy space for best response of player i
        for j = p_min:step:p_max
            p(i) = j;
            % get current response for power, utility and snr
            [cr_power, cr_utility, cr_snr] = utility(i,p,d); % QPSK
%             [cr_power, cr_utility, cr_snr] = u(i,p,d*1000);% FSK

            k = k + 1;

            if cr_utility > br_utility(i)
                br_utility(i) = cr_utility;
                br_power(i) = cr_power;
                br_snr(i) = cr_snr;
            end


            utilityVsPowerTracer(i, ind) = cr_utility;
            ind = ind + 1;


        end
        p(i) = br_power(i);

    end

    powerTracer(:,convergence) = p';

    utilityVsIterationsTracer(:,convergence) = br_utility';

    convergenceTest = true;
    convergenceCount = convergenceCount + 1;
    for k = 1:N

        if (convergence > 1 && powerTracer(k,convergence) - powerTracer(k,
convergence - 1) ~= 0)
```

```
            convergenceTest = false;
        end
        if (powerTracer(k,convergence) == p_max)
            convergenceTest = true;
            ceilingHit = true;
            break;
        end
    end

    if ((convergence > 1 && convergenceTest) || ceilingHit)
        break;
    end
end

if convergenceCount == iterations || ceilingHit
    disp 'Did not converge'
else
    br_power
    convergenceCount
    br_utilitySum = sum(br_utility)
end

%%%%%%%%%%%%%%% CHECK FOR A PARETO SUPERIOR POWER VECTOR %%%%%%%%%%%%%%%%%%

timesParetoFound = 0;
for mu = 1:-0.05:0
    paretoDominant = false;
    paretoRegion = true;
    pp = mu .* p;

    paretoUtilitySum = 0;

    for i = 1:N
        [cr_power, cr_utility, cr_snr] = utility(i,pp,d); % QPSK
%         [cr_power, cr_utility, cr_snr] = u(i,pp,d*1000); % FSK
        paretoUtilitySum = paretoUtilitySum + cr_utility;

        if cr_utility < br_utility(i)
            paretoRegion = false;
            break;
        elseif cr_utility > br_utility(i)
            paretoDominant = true;
        end
    end

    if paretoRegion && paretoDominant
        timesParetoFound = timesParetoFound + 1;

        break;
    end

end

if timesParetoFound > 0
```

```
    disp 'Pareto Dominant power vector found';
    mu
    paretoUtilitySum
end

%%%%%%%%%%%%   PLOT sumUitility Vs mu  %%%%%%% %%%%%%%%%%%%%%%%%%%%%%%%%%%
mu_spacing = 200;
mu = linspace(0, 2, mu_spacing);
paretoUtilitySumTracer = zeros(1, mu_spacing);

for j = 1:mu_spacing
    pp = mu(j) .* p;
    paretoUtilitySum = 0;
    for i = 1:N
        [cr_power, cr_utility, cr_snr] = utility(i,pp,d); % QPSK
%         [cr_power, cr_utility, cr_snr] = u(i,pp,d*1000); % FSK
        paretoUtilitySum = paretoUtilitySum + cr_utility;
    end
    paretoUtilitySumTracer(1, j) = paretoUtilitySum;
end

figure;
plot(mu, paretoUtilitySumTracer);

title('Sum of Utility vs \mu (0 < \mu < 2)', 'FontSize', 15);
ylabel('Utility Sum', 'FontSize', 15);
xlabel('\mu', 'FontSize', 15);


%%%%%%%%%%%%   PLOT POWER OVER THE ITERATIONS %%%%%%%%%%%%%%%%%%%%%%%%%%%

color = jet(N); % for different color of plots for the different players
figure;
hold on;

x_dim = size(powerTracer,2);

for i = 1:N

    plot(1:convergenceCount-1, powerTracer(i,1:convergenceCount-1), 'Color',
color(i,:))
end
hold off;
ylabel('power levels', 'FontSize', 12);
xlabel('iterations', 'FontSize', 12);
title('Convergence to a Nash Equilibrium (Mutual Best Responses)',
'FontSize', 12);
set(gca,'XTick',[1:convergenceCount-1])
grid on;

%%%%%%%%%%%%   PLOT UTILITY OVER THE ITERATIONS
%%%%%%%%%%%%%%%%%%%%%%%%%%%

color = jet(N); % for different color of plots for the different players
```

```
figure;
hold on;

x_dim = size(utilityVsIterationsTracer,2);

for i = 1:N

    plot(1:convergenceCount, utilityVsIterationsTracer(i,1:convergenceCount),
'Color', color(i,:))
end
hold off;

%%%%%%%%%%%  PLOT UTILITY for the last iteration
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure;
hold on;
myMax = 2000;
for i = 1:N
    plot (1:myMax,utilityVsPowerTracer(i,1:myMax), 'Color', color(i,:));
end
ylabel('Utility', 'FontSize', 12);
xlabel('Power (W)', 'FontSize', 12);
% set(gca,'XTick',[1:myMax]/1000)
set(gca,'XTick',1:500:2000)
set(gca,'XTickLabel',[1:500:myMax]/10000)
hold off;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

B2 The Hedge Algorithm (Learning Algorithm)

```
function cumUtil=  utilityComparator2(ne_power,d)
% Hedge algorithm (Freund & Shapire)
% Developed by Oscar Ondeng, University of Nairobi. 22-09-15.


global N;

global p_max; % maximum power transmission level
global step; % the steps in which the power levels can increase
global p_min; % minimum power transmission level


strategy = 0;

learningIterations = 50; % number of games to be played

global cumUtilTrace;
global powerTrace;
cumUtilTrace = zeros(N, learningIterations);
```

```
powerTrace = zeros(N,learningIterations);

global weightsCumUtility; % this is for calculations
weightsCumUtility =  zeros(N, (p_max - p_min) / step + 1);

cumUtil = zeros(1, N); % this is for output
cumPower = zeros(1, N); % useful for finding average power over all
iterations
p = zeros(1, N);

power = zeros(1,N);
util = zeros(1,N);
snr = zeros(1, N);

W = zeros(N, (p_max - p_min) / step + 1); % weights for the updates


% make a number of iterations of play
for i = 1:learningIterations

    % set the strategies of all players before playing the game
    for j = 1:N

        if mod(j,2) == 1

            % for player 1, play based on a learning strategy
            if i > 1
                best_index = find(W(j,:) == max(W(j, :))); % max weight
                p(j) = step * best_index;
            else
                % use ne_power for the first iteration
                p = ne_power; % power at NE
            end
        else

            % for player 2, employ a fixed strategy!
%             p(1, j) = fixedStrategy();
            p(j) = fixedStrategy2(mod(strategy,4) + 1);
            strategy = strategy + 1;
        end
    end

    % once the strategies are set, play the stage game
    for j = 1:N
        % pure strategies
        [power(j), util(j), snr(j)] = utility(j,p,d); % QPSK
    end

    cumUtil = cumUtil + util;
    cumPower = cumPower + p(1, :);

     cumUtilTrace(:,i) = cumUtil';
    powerTrace(:, i) = p(1,:)';
```

```
    % get the weights for the next iteration
    W = getWeights(p, d);

end


avgUtility = cumUtil / learningIterations
avgPower = cumPower / learningIterations

close all

%%%%%%%%%%%%%   PLOT POWER OVER THE ITERATIONS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

color = jet(N); % for different color of plots for the different players
figure;
hold on;

x_dim = size(powerTrace,2);
x = N;
for i = 1:N
    %     plot(1:x_dim, powerTracer(i,:), 'Color', color(i,:))
    subplot(2,1,x); x = x - 1;
    plot(1:learningIterations, powerTrace(i,:), 'Color', color(i,:))
end
hold off;
subplot(2,1,2);
ylabel('power levels', 'FontSize', 12);
xlabel('iterations', 'FontSize', 12);
title('Strategies of the learning player', 'FontSize', 12);
% set(gca,'XTick',1:learningIterations);
grid on;

subplot(2,1,1);
ylabel('power levels', 'FontSize', 12);
xlabel('iterations', 'FontSize', 12);
title('Strategies of the adversarial opponent', 'FontSize', 12);
% set(gca,'XTick',1:learningIterations);
grid on;


%%%%%%%%%%%%%   PLOT CUMULATIVE UTILITIES OVER THE ITERATIONS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

color = jet(N); % for different color of plots for the different players
figure;
hold on;

x_dim = size(cumUtilTrace,2);

for i = 1:N
%     plot(1:x_dim, powerTracer(i,:), 'Color', color(i,:))
plot(1:learningIterations, cumUtilTrace(i,:), 'Color', color(i,:))
end
```

```
hold off;
ylabel('Cumulative Utility', 'FontSize', 12);
xlabel('iterations', 'FontSize', 12);
title('Cumulative Utilities of a learning player and an adversarial
opponent', 'FontSize', 12);
% set(gca,'XTick',1:learningIterations);
grid on;
```

B3 Implementation of Hybrid Play

```
% hybridPlay.m
% Developed by Oscar Ondeng, University of Nairobi. 22-09-15.


clc;
close all;
clear all;
tic
firstTime = toc;

%% initialize variables
global N;
N = 2;  % number of players
maxIterations = 150; %  number of stage games
global iterationsCounter; % useful in regretVector.m

global p_max; global step; global p_min;
p_max = 2; % maximum power transmission level
step = 0.0001; % the steps in which the power levels can increase
p_min = step; % minimum power transmission level

p = ones(1, N) * p_min; % initialize power levels; p => power vector of
current iteration
p_prev = p_min + (p_max - p_min - 1.5).*rand(1, N); % p_prev => power vector
of previous iteration;

d =  [1.2 1.2];

strategy = 0;

 % initialize vectors
 br_power = zeros(1, N); % power at the best response
 br_utility = zeros(1, N); % utility at the best response
 br_snr = zeros(1, N); % snr at the best response

 power = zeros(1,N);
 util = zeros(1,N);
 snr = zeros(1, N);

 cumUtility = zeros(1, N);
 cumPower = zeros(1, N); % useful for finding average power over all
iterations

 global cumUtilTrace;
```

```
 global powerTrace;
 cumUtilTrace = zeros(N, maxIterations);
 powerTrace = zeros(N,maxIterations);

 playModel = 0; % adversay's detected play model
 % 0  => BRP (default); 1 => deterministic; 2 => probabilistic
 global previousPlayModel; % used to detect changes in the play model

% variables pertinent to BRP
 global historySize;
historySize = 1;
global R_history;
% set historySize to -1 to use entire histroy
% set historySize to +ve value to use varying amounts of history
% historySize of 1 => BR response; don't set historySize to 0
R_history = zeros(historySize,N,(p_max - p_min) / step + 1); % set the regret
vector

R = zeros(N,(p_max - p_min) / step + 1); % the regret vector
R_cumulative = zeros(N,(p_max - p_min) / step + 1); % the cumulative regret
vector

% variables pertinent to model of play determiniation
global historySize_strategy; % used in the assessment of adversary's strategy
historySize_strategy = 5;
global P_history;
% set historySize to -1 to use entire histroy
% set historySize to +ve value to use varying amounts of history
% historySize of 1 => BR response; don't set historySize to 0
P_history = zeros(historySize_strategy,N,(p_max - p_min) / step + 1); % set
the regret vector

global BR;
BR = true; % BRP is default

global pointOfChange;
pointOfChange = 0;

% variables pertinent to HA
W = zeros(N, (p_max - p_min) / step + 1); % weights for the updates
global weightsCumUtility; % this is for calculations
weightsCumUtility =  zeros(N, (p_max - p_min) / step + 1);

% variables pertinent to HMA
global history;
history = zeros(N, maxIterations);
global sequenceSize; % size of recent history to use for searching through
the previous histroy
sequenceSize = 4;
global predictedSnr;

% variables pertinent to plotting
playModelTracer = zeros(1, maxIterations);
```

```
 for game = 1:maxIterations
     iterationsCounter = game;
     playModelTracer(game) = playModel;

     % for progress report
     if mod(game, 5) == 0
         timer = toc;
         tic;
         disp(['Running Time at start of iteration ' num2str(game) ': '
num2str(timer/60) ' min']);

     end
     if game >= 40
         disp(['iteration: ' num2str(game)]);
     end

     %% set the strategies of player one and two
     for player = 1:N
         if mod(player, 2) == 1

             % if adversary is employing BRP play model
             if playModel == 0 % BRP
                 % set strategy BRP model of play
                 % get the normalized regret vector
                 total_regret = sum(R_cumulative,2);
                 total_regret = total_regret(player); % pick only the ith
element
                 if total_regret > 0
                     R(player,:) = R_cumulative(player,:) ./ total_regret; %
normalized regret vector for player i
                     % pure strategies
                    best_index = find(R(player,:) == max(R(player,:))); %
best-performing strategy in previous iteration
                    p(1,player) = step * best_index;
                 end

             elseif playModel == 1 % deterministic
                 % set strategy using the HMA
                 p(1, player) = getOptimum(predictedSnr);

             else % probabilistic
                 % set strategy using HA
                 if game > 1
                     best_index = find(W(player,:) == max(W(player, :))); %
max weight
                     p(player) = step * best_index;
                 end
             end

             % set player one's strategy (based on adversary's play model
detected)

         else
             % set player two's strategy
```

```
            if game < 11 % 1 -> 100
                % play BRP strategies
                % get the normalized regret vector
                total_regret = sum(R_cumulative,2);
                total_regret = total_regret(player); % pick only the ith
element
                if total_regret > 0
                    R(player,:) = R_cumulative(player,:) ./ total_regret; %
normalized regret vector for player i
                    % pure strategies
                   best_index = find(R(player,:) == max(R(player,:))); %
best-performing strategy in previous iteration
                   p(1,player) = step * best_index;
                end


            elseif game < 41 % 101 -> 200
                % play deterministic strategies
                p(1, player) = fixedStrategy2(mod(strategy,4) + 1);
                strategy = strategy + 1;
            elseif game < 71 % 201 -> 300
                % play probabilistic strategies
                 p(1, player) = fixedStrategy();
            elseif game < 91
                % play BRP strategies again
                if game == 61
                    R = zeros(N,(p_max - p_min) / step + 1); % the regret
vector
                    R_cumulative = zeros(N,(p_max - p_min) / step + 1); %
the cumulative regret vector
                end
                total_regret = sum(R_cumulative,2);
                total_regret = total_regret(player); % pick only the ith
element
                if total_regret > 0
                    R(player,:) = R_cumulative(player,:) ./ total_regret; %
normalized regret vector for player i
                    % pure strategies
                   best_index = find(R(player,:) == max(R(player,:))); %
best-performing strategy in previous iteration
                   p(1,player) = step * best_index;
                end
            elseif game < 121
                % play deterministic strategies again
                if game == 81
                    strategy = 0;
                end
                p(1, player) = fixedStrategy2(mod(strategy,4) + 1);
                strategy = strategy + 1;
            elseif game
                % play probabilistic strategies again
                 p(1, player) = fixedStrategy();
            end
        end
    end
```

```
    %% play the game
    for player = 1:N
        [power(player), util(player), snr(player)] =
utility(player,p(1,:),d); % QPSK

        history(player, game) = snr(player) / p(1, player); % useful only
for HMA
    end


    %% calculate vectors for next iteration for the learning player(s) and
update variables useful for plotting
    % update regret vector

    % detect adversary's strategy (done only by learning player i.e.
    % player 1)


    previousPlayModel = playModel;
    playModel = getPlayModel(p(1,2)); % input to getPlayModed is player 2's
strategy

%       playModel = getPlayModel_force_HMA;
%       playModel = 2;

    R_cumulative = regretVector(p(1,:),d,R_cumulative, util);

    if playModel == 2
        if previousPlayModel ~= playModel % detect a change in play model
            W = zeros(N, (p_max - p_min) / step + 1); % weights for the
updates
            weightsCumUtility =  zeros(N, (p_max - p_min) / step + 1);
        end
        W = getWeights(p, d);

    end



    %%% update variables useful for plotting
    cumUtility = cumUtility + util;
    cumPower = cumPower + p(1, :);

    cumUtilTrace(:,game) = cumUtility';
    powerTrace(:, game) = p(1,:)';

 end

 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
cumUtility
avgUtility = cumUtility / maxIterations
avgPower = cumPower / maxIterations
```

```
%%%%%%%%%%%%%   PLOT POWER OVER THE ITERATIONS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

color = jet(N); % for different color of plots for the different players
figure;
hold on;

x_dim = size(powerTrace,2);
x = N;
for i = 1:N
    %     plot(1:x_dim, powerTracer(i,:), 'Color', color(i,:))
    subplot(2,1,x); x = x - 1;
    plot(1:maxIterations, powerTrace(i,:), 'Color', color(i,:))
end

hold off;

subplot(2,1,2);
ylabel('power levels', 'FontSize', 12);
xlabel('iterations', 'FontSize', 12);
title('Strategies of the learning player', 'FontSize', 12);
set(gca,'XTick',0:10:150)
grid on;

subplot(2,1,1);
ylabel('power levels', 'FontSize', 12);
xlabel('iterations', 'FontSize', 12);
title('Strategies of the adversarial opponent', 'FontSize', 12);
set(gca,'XTick',0:10:150)
grid on;

figure;
plot(playModelTracer);
grid on;

disp(['Total Running Time: ' num2str((toc - firstTime) / 60) ' min']);
```

B4 Function Used by the Hybrid Implementation

```
function playModel = getPlayModel(p)
% function determines the adversarial player's mode of play
% (only one adversary)
% function used in conjuction with hybridPlay.m
% Developed by Oscar Ondeng, University of Nairobi. 22-09-15.



%% initialize variables
global historySize_strategy;
global P_history;
global iterationsCounter;

global previousPlayModel;

global history; global sequenceSize; global BR;
global predictedSnr;

global pointOfChange;

variationThreshold = 0.3; % the amount of variation in Watts in the power
correlationThreshold = 0.99999999999;

if iterationsCounter < historySize_strategy
    playModel = 0; % without sufficient history, BRP is default
    return;
end

%% determine whether adversary is playing BR
% by assessing the variation in the last few iterations

newP = mod(iterationsCounter - 1, historySize_strategy) + 1;
P_history(newP) = p;

if max(P_history) - min(P_history) < variationThreshold
    playModel = 0;
    BR = true;
else % adversary not playing BR
    % determine whether adversary is playing a deterministic model or
probabilisitic model

    if previousPlayModel == 0 && BR % a change in play model from BR has been
detected
        pointOfChange = iterationsCounter;
        BR = false;
    end
    % determine xcorr of pattern with history of interest
    if iterationsCounter  > pointOfChange + sequenceSize * 2
        j = 1; % player one
        i = iterationsCounter;
```

```
        % perform xcorrelation
        pattern = history(j, i + 1 - sequenceSize : i);
        history_to_check = history(j, pointOfChange : i - 1);

        xcor = xcorr(history_to_check, pattern);
        perf_xcor = xcorr(pattern, 0);
        xcor = xcor ./ perf_xcor;
        xcor(xcor > 1.00001) = 0;

        xcor = xcor(size(history_to_check,2) - sequenceSize + 1 : end);

        indexMax = find(xcor == max(xcor));
        selectedIndex = indexMax(end);

        if max(xcor) > correlationThreshold && selectedIndex <
size(history_to_check, 2)
            % adversary has deterministic model of play
            playModel = 1;
            predictedSnr = history_to_check(j, selectedIndex + 1);
        else
            % adversary has probabilistic model of play
            playModel = 2;
        end

    else
        playModel = 0; % play BR until there is sufficient history for HMA
    end

end
```

B5. Finding SINR Numerically

```
function sinr = findSINR(M)
% finding the equilibrium snr numerically
% Developed by Oscar Ondeng, University of Nairobi. 22-09-15.

% M is the packet length

accuracy = 0.001;
threshold = 0.001;

for snr = 1:accuracy:20

      diff = snr * M * ((erf(sqrt(snr))) ^ (M - 1)) * (exp(-snr) * snr ^ (-1
      / 2)) / sqrt(pi) - (erf(sqrt(snr))) ^ M;
```

```
    if diff < threshold
        break;
    end
end
```

B6. Function to calculate the utility

```
% utility.m
% Developed by Oscar Ondeng, University of Nairobi. 22-09-15.

% utility function in the game-theoretic power control
% WCDMA

function [br_power, br_utility, br_snr] = utility(i, p, d) %#codegen

% p is the vector of powers of the players
% d is the vector of distances of the players from the base station

% initalize constants
% based on typical values used in a WCDMA network with SF = 256
global L; global M; global R; global W; global noise;
L = 100; % number of information bits/frame with 1/3 rate coding
M = 150; % total number of bits/frame
R = 15e3; % bit rate
W = 5e6; % Spread Spectrum Bandwidth
noise = (4e-21) * W; % noise power = noise floor * bandwidth

% path loss
h = pathLoss(d); % path loss in decibels; using Extended HATA
h = 1 ./ (10 .^ (h./10)); % path loss as a ratio

% vector of powers other than for player i
p_i = p;
p_i(i) = 0;

% signal to noise ratio
snr = W / R * (p(i) * h(i)) / (p_i * h' + noise);

% symbol and bit error rate (for QPSK modulation)
ser = erfc(sqrt(snr)) - (1/4)*(erfc(sqrt(snr))).^2; % revised ser, based on
Proakis...

ber = ser / 2; %QPSK

% ber = 0.5 * exp(-0.5 * snr); %FSK

% power of zero yields utility of zero
if p(i) == 0
    u = 0;
else
    u = L * R * ((1 - 2 * ber) ^ M) / (M * p(i));
end
```

```
br_power = p(i);
br_utility = u;
br_snr = snr;
```

APPENDIX C

Publications Resulting from this Research

i. **"Game-Theoretic Transmit-Power Control in Cognitive Radio with Pareto-Improvement of the Nash Equilibrium", International Journal of Scientific and Engineering Research, Vol. 6, No. 4, April, 2015**

Abstract – In implementing cognitive radio networks, transmit-power control is one of the key tasks of the cognitive cycle and plays a big role in carrying out spectrum sharing. In this work the transmit-power control of a CDMA cognitive radio network is modeled as a non-cooperative game-theoretic problem. The iterative water filling algorithm is implemented using the best response to the previous play in an attempt to arrive at the Nash Equilibrium. The characteristics of the convergence and of the Nash Equilibrium are studied and of special interest is the Pareto optimality. It is found that the Nash Equilibrium is not Pareto-optimal and a method is proposed and implemented to achieve a power vector which is Pareto-superior to the power vector of the Nash Equilibrium and which yields a higher utility.

Keywords: Cognitive Radio, Nash Equilibrium, Non-Cooperative Game Theory, Pareto Efficiency, Transmit-Power Control

ii. **"Distributed Transmit-Power Control in Cognitive Radio Networks Using a Hybrid-Adaptive Game-Theoretic Technique", IEEE Africon Proceedings, September, 2015**

Abstract—This paper studies game-theoretic distributed transmit-power control in a cognitive radio network. It presents a hybrid-adaptive algorithm that interfaces Iterative Water-Filling with two learning algorithms: the Hedging Algorithm and the Historical Matching Algorithm. Iterative Water-Filling helps achieve a fast convergence whereas the learning algorithms help guard against exploitation. The learning algorithms employed are selected based on their performance in deterministic and probabilistic network environments. The hybrid-adaptive algorithm is shown to offer improvements on other methods published. It also performs better than Iterative Water-Filling and the learning algorithms taken in isolation. The main metric is the utility achieved by the players in the game-theoretic setting.

Keywords—Cognitive radio, game theory, iterative water-filling, learning algorithms, Nash Equilibrium, transmit-power control.