

Game Theoretic Methods for Networked Sensors and Dynamic Spectrum Access

by

Michael Maskery

B.Sc., Queen's University, 2000

M.Sc., The University of Ottawa, 2003

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

The Faculty of Graduate Studies

(Electrical and Computer Engineering)

The University Of British Columbia

December, 2007

© Michael Maskery 2007

Abstract

Automated devices enabled by wireless communications are deployed for a variety of purposes. As they become more ubiquitous, their interaction becomes increasingly important for coexistence when sharing a scarce resource, and for leveraging potential cooperation to achieve larger design goals. This thesis investigates the use of game theory as a tool for design and analysis of networked systems of automated devices in the areas of naval defence, wireless environmental monitoring through sensor networks, and cognitive radio wireless communications.

In the first part, decentralized operation of naval platforms deploying electronic counter-measures against missile threats is studied. The problem is formulated as a stochastic game in which platforms independently plan and execute dynamic strategies to defeat threats in two situations: where coordination is impossible due to lack of communications, and where platforms hold different objectives but can coordinate, according to the military doctrine of Network Enabled Operations. The result is a flexible, robust model for missile deflection for advanced naval groups.

Next, the problem of cooperative environmental monitoring and communication in energy-constrained wireless sensor networks is considered from a game-theoretic perspective. This leads to novel protocols in which sensors cooperatively trade off performance with energy consumption with low communication and complexity overhead. Two key results are an on-line adaptive learning algorithm for tracking the correlated equilibrium set of a slowly varying sensor deployment game, and an analysis of the equilibrium properties of threshold policies in a game with noisy, correlated measurements.

Finally, the problem of dynamic spectrum access for systems of cognitive radios is considered. A game theoretic formulation leads to a scheme for competitive bandwidth allocation which respects radios' individual interests while enforcing fairness between users. An on-line adaptive learning scheme is again proposed for negotiating fair, equilibrium resource allocations, while dynamically adjusting to changing conditions.

Table of Contents

Abstract	ii
Table of Contents	iii
List of Tables	viii
List of Figures	ix
Acknowledgements	xi
Dedication	xii
Statement of Co-Authorship	xiii
1 Introduction	1
1.1 Principles of Game Theoretic Design and Analysis	2
1.1.1 Game Theoretic Equilibrium	3
1.1.2 Stochastic Games	4
1.1.3 Benefits and Challenges of Game-Theoretic Design and Analysis	5
1.2 Main Contributions	8
1.2.1 Contributions to Decentralized Force Protection	8
1.2.2 Contributions to Decentralized Sensor Activation Control	10
1.2.3 Contributions to Decentralized Dynamic Spectrum Access	11
1.2.4 Contributions to Self-Configuration of Sensors with Global Games	11
1.3 Review of Technology	12
1.3.1 Missile Deflection and Network Centric Warfare	12
1.3.2 Wireless Sensor Networks	13
1.3.3 Cognitive Radio Dynamic Spectrum Access	14
1.4 Contents of this Thesis	15

Table of Contents

Bibliography	19
2 Decentralized Force Protection: Nash Equilibrium	24
2.1 Introduction	24
2.2 Motivation and Review of Stochastic Games	26
2.2.1 Decentralized Architecture for Implementing Countermeasures with a Common Air Picture	27
2.2.2 Motivation: Pulse Sequence Jamming over a Link 16 Network	27
2.2.3 Review of Stochastic Shortest Path Games	29
2.3 Decentralized Missile Deflection as a Stochastic Shortest Path Game	32
2.3.1 State Space of the Multiple Missile Threat	34
2.3.2 Dynamics of a Radar-Guided Multiple Missile Threat	35
2.3.3 Countermeasure Actions	38
2.3.4 Platform Objectives and State Costs	39
2.4 Computation of Decentralized Missile Deflection Policies	44
2.4.1 Computation of an Optimal Response for a Single Ship	44
2.4.2 Exploiting Transient Classes in the MDP	47
2.4.3 Nash Equilibrium Solution for Multiple Ships	50
2.4.4 Pareto Optimality and Nash Equilibrium	52
2.5 Introducing CM Limitations via Constraints	53
2.5.1 Limiting CM availability via State Space Augmentation	54
2.5.2 Limiting CM Directly in the Linear Program	55
2.5.3 Constraining CM Sequences	57
2.6 Numerical Results	58
2.6.1 Maximizing Minimum Approach of a Missile with Simple Targeting	58
2.6.2 Minimizing Probability of Hit for a Missile with Forward Momentum	60
2.7 Conclusion	68
Bibliography	71
3 Decentralized Force Protection: Correlated Equilibrium	73
3.1 Introduction	73
3.1.1 Network Enabled Operations	76
3.1.2 The Case for Correlated Equilibrium in NEOps	78
3.2 Network Enabled Operations in Dynamic Missile Deflection	82
3.2.1 Single Missile Process	83

Table of Contents

3.2.2	Multiple Missile Process	85
3.3	Decentralized Missile Deflection as a Transient Stochastic Game	86
3.4	Characterization of the NEOps Equilibrium Strategy	88
3.4.1	Correlated Equilibrium as the Solution to a Bilinear Program	92
3.5	Information Theoretic Metrics in NEOps	95
3.5.1	Optimizing Information Flow in NEOps	95
3.5.2	Implementation Architectures for NEOps Equilibrium Strategies	99
3.6	Numerical Example of NEOps	104
3.6.1	Inner Game (Single Missile in Attack Mode)	105
3.6.2	Outer Game (Selection from Three Missiles)	109
3.6.3	Outer Game (Myopic Selection from Ten Missiles)	111
3.7	Conclusion	113
Bibliography		115
4	Decentralized UGSN Activation Control	118
4.1	Introduction	118
4.2	Decentralized Energy-Aware Node Activation	121
4.2.1	Measurement Model	122
4.2.2	Activation Control Utility Function	122
4.2.3	Nonlinear Adaptive Filtering for Sensor Activation	124
4.2.4	Performance of Regret Tracking	128
4.3	Correlated Equilibrium and Convergence	129
4.3.1	Definition of Correlated Equilibrium	129
4.3.2	Convergence Analysis of Algorithm 4.2.2	131
4.3.3	Tracking Performance of Algorithm 4.2.2	134
4.3.4	Algorithm Improvements and Pricing	136
4.4	Test Bed and Numerical Results for ZigBee-enabled UGSN	138
4.4.1	Tracking Dynamic Targets: Comparison of Performance	140
4.4.2	Parameter Selection and Sensor Estimation	142
4.5	Discussion	144
Bibliography		147
5	Game Theoretic Dynamic Spectrum Access	150
5.1	Introduction	150

Table of Contents

5.2	Opportunistic Spectrum Access Model	153
5.3	Estimating Channel Characteristics	154
5.3.1	Channel Contention Estimate	154
5.3.2	Other Channel Characteristics	158
5.4	System Performance and Radio Utility	159
5.4.1	Global System Utility	159
5.4.2	Local Radio Utility	159
5.5	Decentralized Adaptive Channel Allocation	161
5.5.1	Convergence of Adaptive Channel Allocation	163
5.5.2	Other Adaptive Strategies	165
5.6	Numerical and Simulation Results	166
5.6.1	Effect of Utility Pricing on Performance of Regret Tracking	166
5.6.2	Comparison of Algorithms in a Static Environment	168
5.6.3	Performance of Regret Tracking in a Dynamic Environment	171
5.7	Conclusions	171
Bibliography		173
6	Self-Configuration of Sensors: Global Games	176
6.1	Introduction	176
6.2	A Global Game for Sensor Mode Selection	179
6.2.1	Characteristics of Global Games	179
6.2.2	Model Description	180
6.2.3	Predicting the Proportion of Active Sensors	183
6.3	Nash Equilibrium Threshold Strategies	185
6.3.1	Nash Equilibrium Threshold Strategies for Uniform Observation Noise 187	
6.3.2	Threshold Strategies for Gaussian Priors and Noise	190
6.4	Further Analysis of Mode Selection Thresholds	191
6.4.1	Adaptive Self-Configuration of Mode Selection Thresholds	192
6.4.2	Robustness of Threshold under Measurement Averaging	193
6.4.3	Robustness of Threshold under Variance-based Measurement Quality	196
6.5	Numerical Examples	197
6.5.1	Single Utility, Weak Congestion	197
6.5.2	Multiple Utilities, Mixed Congestion	200
6.5.3	Convergence of Algorithm 6.4.1	203

Table of Contents

6.6	Conclusions	204
Bibliography		206
7	Conclusion	208
7.1	Common Approach to Game Theoretic Design and Analysis	208
7.1.1	System Objectives and Capabilities	208
7.1.2	Component Awareness	210
7.1.3	Local Incentives	211
7.1.4	Equilibrium Analysis	213
7.2	Evaluation of Results	216
7.2.1	Analytical Tools Used	216
7.2.2	Strengths and Weaknesses	217
7.2.3	Characteristics of Appropriate Applications	220
7.3	Future Work	222
Bibliography		225
 Appendices		
A	UGSN Architecture	227
A.1	Sensor Network Model and Architecture	227
A.2	ZigBee/IEEE 802.15.4 Communication Model	228
Bibliography		233
B	Global Games Proofs	234
Bibliography		239

List of Tables

3.1	Missile hit probabilities	107
3.2	Missile deflection game costs	111
4.1	Simulation Parameters	139
4.2	Regret-based vs. greedy sensor activation	142
A.1	IEEE 802.15.4 Standard Values	230
A.2	IEEE 802.15.4 Power Consumption	231

List of Figures

1.1	Organization of Thesis.	16
2.1	Overview of decentralized missile deflection system	28
2.2	Missile transition probabilities for example 1	59
2.3	Expected Nash equilibrium costs for example 1	61
2.4	Expected Pareto optimal costs for example 1	62
2.5	Sample path of Nash equilibrium for missile deflection example 1	63
2.6	Missile transition probabilities for example 2	65
2.7	Sample path of Nash equilibrium for missile deflection example 2	65
2.8	Performance of Nash equilibrium for missile deflection	67
3.1	Decentralized missile deflection using NEOps	77
3.2	Communication architecture for NEOps	100
3.3	Missile transition probabilities for NEOps example	106
3.4	Mutual Information for NEOps example	108
4.1	Activation control as adaptive filtering	123
4.2	Sensor utility and target tracking	140
4.3	Sensor lock time and target tracking	141
4.4	Sensor network pricing performance	143
4.5	Sensor network timer performance	145
5.1	Block diagram of cognitive radio system.	155
5.2	Performance of contention estimate for cognitive radio.	157
5.3	Cognitive radio performance vs. utility parameters	167
5.4	Performance of cognitive radio decentralized channel allocation	169
5.5	Regret analysis of cognitive radio decentralized channel allocation	170
5.6	Performance of cognitive radio channel allocation for dynamic conditions . .	172

List of Figures

6.1	Block diagram for global game	184
6.2	Global game example 1: sensor utility	198
6.3	Global game example 1: threshold vs. noise	199
6.4	Global game example 1: thresholds for diverse sensors	200
6.5	Global game example 2: sensor utility	201
6.6	Global game example 2: equilibrium conditions	202
6.7	Convergence of iterative threshold computation	204
A.1	IEEE 802.15.4 Transmission Energy	231
A.2	IEEE 802.15.4 Success Probability	232

Acknowledgements

This work would not have been possible without the support, enthusiasm and guidance of my supervisor, Vikram Krishnamurthy, to whom I owe a debt of thanks. I would also like to acknowledge the support and feedback from my fellow graduate students at UBC, and from the scholars who visited UBC or whom I met while working on this material. Finally I would like to thank Christina O'Regan, Qing Zhao and George Yin, dedicated researchers and scholars with whom I collaborated to produce the manuscripts contained herein.

The work produced here was funded in part by Defence Research and Development Canada, and by NSERC Postgraduate Doctoral Scholarship.

Dedicated to my family, my friends, and to Jacqueline.

Statement of Co-Authorship

This thesis is based on several manuscripts, resulting from collaboration between multiple researchers. A version of Chapter 2 will appear in 2007 in IEEE Transactions on Aerospace and Electronics Systems. This was co-authored with Vikram Krishnamurthy and Christina O'Regan. A version of Chapter 3 will also appear in 2007 in IEEE Transactions on Aerospace and Electronics Systems, and was co-authored with Vikram Krishnamurthy. A version of Chapter 4 was submitted to IEEE Transactions on Signal Processing. This was co-authored with Vikram Krishnamurthy and George Yin. A version of Chapter 5 was submitted to IEEE Transactions on Communications. This was co-authored with Vikram Krishnamurthy and Qing Zhao. A version of Chapter 6 was submitted to IEEE Transactions on Signal Processing. This was co-authored with Vikram Krishnamurthy. The author's specific contributions in these collaborations are detailed below.

- Identification and design of research program: The research program was designed jointly based on ongoing discussion between the author and Vikram Krishnamurthy. Identification and design in specific research topics also depended on significant input from collaborators as follows:
 - The overall problem formulation and physical scenario for decentralized missile deflection in Chapters 2 and 3 depended on input from Christina O'Regan and other researchers at Defence Research and Development, Canada.
 - The results on strong and weak convergence of the regret-based learning algorithms in Chapters 4 and 5 are due in part to George Yin.
 - The formulation of the cognitive radio model in Chapter 5 was performed in collaboration with Qing Zhao.
- Performing the research: All major research, including detailed problem specification, model design, performance of analysis and identification of results was performed by the author, with assistance from Vikram Krishnamurthy.

Statement of Co-Authorship

- Data analyses: All numerical examples, simulation, and data analysis were performed by the author.
- Manuscript preparation: The author prepared the majority of all manuscripts, with the exception of the following:
 - Qing Zhao wrote the section on related work in the introduction of Chapter 5, and was responsible for suggestions and improvements throughout the chapter.
 - George Yin was responsible for the technical treatment of convergence and tracking for the regret-based learning algorithms in Chapter 4.
 - Vikram Krishnamurthy outlined the proposed extensions in the conclusion of Chapter 4. He was also responsible for various suggestions and improvements throughout this thesis.
 - Christina O'Regan and Pierre Lavoie were responsible for editing and suggestions in Chapters 2 and 3.

Chapter 1

Introduction

Since the first microcontroller was released in 1976, embedded electronics have become an integral part of modern civilization. Even today, some thirty years later, new uses are still being found for embedded electronics, which interact ever more seamlessly with their environment and their human users.

The Case for Inter-Device Interaction: With electronics embedded everywhere, it is unavoidable that embedded devices will encounter one another as they perform their core functions. The prevalence of these interactions is compounded by the recent trend of incorporating wireless communications into devices to enhance their functionality. Indeed, the wireless environment is inherently far-reaching, linking together devices that may be separated by large physical distances. This thesis comprises a study of the interaction among such “networked” devices, which must coordinate and coexist in this wireless landscape. In particular, interaction is studied in networks of wireless-enabled missile deflection systems, wireless sensor networks for environmental monitoring, and cognitive radio wireless communication networks.

As each device we consider is an independent entity, these networks require new tools for engineering design and analysis. For example, although devices might be capable of wireless communication, the bandwidth originally allocated for coordination may be co-opted to satisfy other, more pressing needs. In addition, decentralization may be required to support practical features such as self-organization and robustness, since the device network may be too complex or dispersed to coordinate centrally. Consequently, devices comprising modern systems may exhibit a high level of autonomy, and may need to operate based only on what local information can be easily gathered.

Decentralized Systems Approach: Since the independently operating devices considered in this thesis interact, they may be considered as a group to comprise a decentralized system, which can be analyzed for global patterns of behaviour. One natural and emerging theoretical framework for studying the behaviour of such “multi-agent systems” [55] is

game theory, and indeed the major theme of this thesis is the use of noncooperative game theory in a *cooperative* way, to analyze independent and locally aware devices in a decentralized system set up to achieve desirable overall behaviour. Two important tasks must be completed for a proper game theoretic study. First, the individual device incentives must be identified, and analyzed to determine whether they favour a fair balance between the device's own objectives and the objectives of others (or the system). This is the problem of determining an appropriate *utility function* for each device, mapping its awareness and action to a real value. Second, recognizing that awareness may be limited and incentives may differ, a mechanism must be found which allows devices to adapt to each other, such that their combined effort reaches a stable operating point, in which each device's actions are an optimal response (from their perspective) to their perceived environment and the actions of others. This is the problem of specifying a mechanism for action that leads to game theoretic *equilibrium*.

The rest of this introductory chapter provides a brief overview of the proposed game theoretic approach to decentralized systems, along with the various applications appearing in this thesis. Since subsequent chapters are intended to be self-contained, more detailed literature reviews, motivations, and descriptions of each topic are contained within each chapter. Notation is also self-contained within each chapter.

1.1 Principles of Game Theoretic Design and Analysis

Game theory is a *complexity-based* theory, along with percolation theory, cellular automata and some aspects of advanced physics and ecology modeling. Such theories derive rich dynamics through the interaction of simple components, and are used either as descriptive tools, to predict the outcome of complex interactions, or as prescriptive tools, to design systems around given interaction rules. Game theory is used for both design and analysis in this thesis, to contribute to a new, rich theory for decentralized systems engineering.

The essential goal of game theory is to describe the “optimal” outcome of a process involving multiple decision makers, or players. Each player's outcome is affected not only by their own decision, but also by the decisions of others. Solving the game, therefore, involves a complex process of players guessing what each other will do, while revealing as little information as possible about their own strategies. Canonical examples are numerous, including the famous “Prisoner's Dilemma,” strictly competitive games, such as “Rock Paper Scissors,” coordination games, and mixed games [25].

Tools of Game Theory: The basic game theoretic approach in this research is well documented in the mathematics, economics and engineering literature. Comprehensive texts on the subject include [25, 46]. The theory was originally developed as a tool for studying spontaneous organization of behaviour among self-interested agents in an interactive environment by mathematicians and economists, notably von Neumann and Morgenstern [56]. Each player is treated as a decision maker with a set of possible actions and a utility for each action, which is to be maximized. The utility, however, is also a function of others' actions, which makes the maximization problem nontrivial.

1.1.1 Game Theoretic Equilibrium

The central concept in noncooperative game theory is the equilibrium, which represents a joint strategy for selecting actions such that no agent can improve their utility by deviating from their part of the strategy. Several variations on the equilibrium concept stand out, including the Nash [44], correlated [5, 6], and communication equilibria [23, 29]. Each variation has its own place, and both the Nash and correlated equilibria are central to this thesis.¹ The Nash equilibrium describes situations where each player must select their actions independently; the covariance between the random variables representing individual player actions is zero. The Correlated equilibrium relaxes this condition, allowing the nonzero covariance and consequently players select actions based on some common information. Several mechanisms have been proposed to realize the correlated equilibrium, using either a central authority who recommends or verifies proposed actions [5, 25, 37], or using a verification scheme among the players [7, 9, 10, 22, 27, 54, 60].

Relationship between Nash and Correlated Equilibrium: The correlated equilibrium is an interesting generalization of the Nash equilibrium, and we study it extensively in Chapters 3, 4 and 5. In fact the correlated equilibria comprise a convex polytope in the space of all randomized strategies, with Nash equilibria as extreme points [45]. Consequently, correlated equilibria are particularly simple to compute, as they are solutions to a linear feasibility problem, or a linear programming problem if one seeks to find the “best” correlated equilibrium according to some linear criteria.

Learning in Games: An equilibrium strategy is a probability distribution over the joint actions of all players. However, in practical situations awareness is often distributed such

¹The communication equilibrium is a straightforward extension to the correlated equilibrium, but typically requires more coordination than is allowed for in the applications considered here.

that no player can possibly compute such an equilibrium on their own. In this case, researchers have proposed adaptive procedures to approximate the equilibrium, with each player learning their part of the equilibrium by repeated interactive play and strategy adjustment. The most well-known such procedure, fictitious play, was introduced in 1951 [48], and has been extensively studied, see [24]. More recently, generalizations have been proposed, in the form of regret-based learning [11, 16, 30, 31] etc., and it is these that are the focus of our work. We use tools from stochastic approximation [36] to further modify these regret-based procedures to an adaptive version, which is more useful in practical applications. Game theory can also be interpreted as an analytical tool for determining the effectiveness and stability of simple, adaptive learning rules. This is the approach taken in Chapters 4 and 5, which sets up rules of behaviour, then analyzes them to show convergence to equilibrium, implying competitive optimality between devices.

1.1.2 Stochastic Games

Traditional game theory addresses a static decision problem. However, many practical problems require a *sequence* of decisions, each of which affects future performance. The problem of trading off present gain and future reward is captured, for a single decision maker, by the well-known Hamilton-Jacobi-Bellman equation of dynamic programming and optimal control theory [13, 47]. The analogous dynamic programming model for multiple decision makers is the subject of the theory of stochastic games [21, 51], which, interestingly, was first published slightly *before* the famous single-decision maker theory of Bellman.

In a stochastic game, multiple players face a sequence of static noncooperative games according to a Markov chain. Each joint action provides some reward (or cost) to each player, and determines a probability distribution, according to which the next game is selected. Standard practice is to formulate and simultaneously solve a number of coupled “stage games” which give the expected total reward associated with each game and action, given the expected actions in the other games. These stage games can be formulated for discounted, finite-horizon, and average cost variants and the equilibrium concept extended to hold for each stage game simultaneously.

Solving Stochastic Games: Solutions of stochastic games are extremely complex and generally less well understood than their static counterparts. For special, simplified cases, including zero-sum games, several methods for computing a stationary Nash equilibrium based on linear programming are proposed in [21], and a homotopy method is proposed in [33]. In Chapters 2 and 3, we propose other computational methods, including a sequence of

linear programs which converges for “team” games, and a bilinear programming formulation for computing correlated equilibria in general stochastic games.

1.1.3 Benefits and Challenges of Game-Theoretic Design and Analysis

The game theoretic approach has several appealing features for system design and analysis. Simple devices with limited awareness can be equipped with pre-configured or configurable utility functions and routines for maximizing their utility in an interactive, even unknown environment. Such devices can then be deployed to organize themselves effectively in a dynamic and unknown environment. As long as the utility functions are properly specified, these “networked” devices can be made to exhibit many desired behaviors.

This game theoretic design approach echoes natural systems in a form of biomimicry [12]; biological agents such as insects have long since evolved the proper procedures and utility preferences to accomplish collective tasks. The same can be said of humans, who orient their utilities according to economics; the proper specification of utilities in this case are dictated by the structure of the economic system and realized through such mechanisms as pricing.

The self-organizing feature of game-theoretic (decentralized) systems results in a specific set of benefits and challenges. While centralized systems can achieve better overall performance when the application is known and information can be easily distributed, there is a growing interest in developing decentralized systems for situations where this is not the case. By spreading intelligence throughout the system, game theoretic systems can eliminate the need to transport information to and from a central point, while still allowing local information exchange to any desired degree.

Benefits: The key benefits of game theoretic design and analysis are listed as follows:

- **Robustness:** Since each device in a decentralized system is independent, the failure of a few devices does not necessarily result in the failure of the entire system. This allows systems to continue functioning in difficult environments, where devices may be maliciously disabled, temporarily cut off from communication, or drained of energy reserves.
- **Scalability:** Likewise, game theoretic systems naturally support the addition of devices with little impact on performance. Moreover, the complexity of each device does not increase with the number of devices in the system, whereas in centralized systems complexity may increase rapidly.

- **Flexible Self-Organization:** Since awareness and decision capabilities are co-located, devices in a decentralized system naturally and independently optimize their own behaviour without outside assistance. Since device awareness includes the actions (or their effects) of others, this results in a spontaneous self-organizing behaviour for the system, even in unknown environments. Moreover, awareness and decisions are local in nature, making the overall complexity of the environment irrelevant to the reaction of the system. This feature also facilitates quick response to local environmental changes, since new information is adapted to at its source.
- **Low Communication Overhead:** The self-organization feature eliminates the need for awareness and decision data to be routed through the system to a central point, thus reducing overhead. However, since the end goal may be to report data, communication may still be needed for transferring (aggregated) information through the system. There is also the possibility of “tuning” system parameters on a slow time scale, which requires low but nonzero bandwidth. A final overhead reduction arises since it is not necessary for mutual awareness to be achieved through explicit communication; devices may simply derive mutual awareness from their observation of their “neighbour’s” observable impact on the immediate environment (e.g. wireless MAC interaction in Chapters 4 and 5).
- **Modular Analysis:** Game theory is a beneficial analytical tool, since it allows for a decomposition of a large-scale problem into a number of sub-problems. Specifically, one can find an equilibrium operating point by considering the optimization of each component (player) in turn, assuming the others are acting optimally. (This is similar to dynamic programming, in which one chooses an optimal action assuming optimal actions will be taken in the future.) This replaces the complex problem of global optimization with a number of smaller optimizations, which are solved simultaneously to provide an equilibrium point, representing a competitive version of optimality. Such an approach is often useful from a computational perspective, and also provides insight into the fundamental mechanics of the problem at hand.

Challenges: The main drawback of game theoretic systems derives from the lack of consensus and coordination among system devices. This gives rise to some key challenges, listed as follows:

- **Limited Awareness:** A typical device in a game theoretic system will only be aware of its immediate environment. (If it were aware of everything, a centralized system

would also be feasible.) Thus, the system often behaves in a myopic fashion, since distant conditions cannot be anticipated.

- **Limited Coordination:** Similarly, only the behaviour of other nearby devices can be observed, and often not directly. Moreover, there is a coupled prediction problem when devices repeat actions over time; to work well together, each device must predict the actions of its neighbours.
- **Limited Intelligence:** Because there are a potentially large number of devices, financial constraints in manufacturing may become involved, making it infeasible to equip system devices with the best computing technology. Much of the time, only very simple algorithms can be realistically implemented.
- **Analytical Assumptions:** Game theoretic analysis requires certain assumptions such as common knowledge of rationality [25] to justify the use of an equilibrium in games involving human players. For analysis of automated devices, this translates into a requirement for common knowledge of rationality for the device *designers*. That is, each device optimizes its own behaviour, based on the assumption that its counterparts operate similarly, and that this fact is common knowledge. In real systems, it may often be difficult to find a model utility function that fits the problem at hand under these assumptions. This is because many systems use a combination of rational behaviour, along with irrational behaviour in the form of automaton-like stimulus/response mappings. Moreover, if many different types of devices interact, the analyst must provide a rational “player” model for each one, a daunting task for open systems.

These challenges lead to suboptimal performance, or ill-fitting analytical models. Performance issues may be partially overcome by choosing a utility function at each sensor which makes the best use of the information at hand, and by choosing a mechanism which quickly leads to coordinated joint behaviour. Although the decentralized nature of awareness and action generally keeps optimality out of reach, this is often more than compensated by the benefits described above. Analytical issues must be overcome by an increase in complexity, by considering multiple game scenarios, combining players, or using hybrid analysis techniques.

1.2 Main Contributions

In this section, we briefly describe the major novel contributions of the chapters comprising this thesis. These contributions fall into two general categories: novel modeling approaches, and novel analytical approaches applied to the game theoretic models defined in each chapter. In this section these contributions are reviewed below, for the reader's convenience, according to the chapters that comprise the body of this thesis. A more detailed description of the contribution of each chapter and related work is described in individual chapters.

1.2.1 Contributions to Decentralized Force Protection

Chapter 2 introduces the idea of game theory as a tool to analyze the problem of decentralized force protection among Naval platforms. Although coordinated force protection (wherein multiple Naval platforms communicate in real time to cooperatively defend against missile threats) is a known problem, such systems are vulnerable to communications failures, which often intentionally accompany a missile attack. Therefore, there exists a need for solution techniques in scenarios when communication is compromised, which is addressed by the game theoretic approach of this chapter. This allows for consideration of platform utilizing pre-computed defensive policies for automatic deployment, which can be implemented in the absence of coordination capabilities. Decentralized force protection is a novel area of research, as inspired by recent interest in network centric warfare. The main contributions of this chapter are:

- A novel formulation of the problem of decentralized force protection against anti-ship missiles using the theory of stochastic games, such that game players (countermeasure-equipped ships) are not strictly adversaries, but rather cooperative, yet independent entities.
- Specification of appropriate defensive objectives for the game, either to maximize the minimum distance of approach of a missile, or to minimize the expected number of missile hits, based on formulation of a shortest path (transient) stochastic game theory.
- Augmentation of the state space of the stochastic game to capture the objective of maximizing the minimum distance of approach, along with a method of exploiting the resulting evolution of state trajectories so as to improve efficiency in computing the solution via linear programming techniques.

- A Nash equilibrium solution based on simple yet effective algorithms for solving Markovian games, representing the best available solution when communication between ships is infeasible.
- Examples, with illustration of the sensitivity of the solution to changes in the missile dynamics.

Chapter 3 extends Chapter 2 by considering the case in which Naval platforms are now capable of communicating while responding to missile threats. However, the recently proposed doctrines of Network Centric Warfare [4] and Network Enabled Operations (NEOps) [8] specify that each participating Naval platform be *self-empowered*, that is, an independent decision maker which may possess its own objective, possibly due to unique local conditions or information at each platform. Since there is currently no encompassing theoretical framework for considering such situations, we propose a game theoretic approach incorporating the concept of a correlated equilibrium, which allows game players to coordinate while still retaining their autonomy. Moreover, this correlated equilibrium concept is inserted into the stochastic game framework in a more definite manner than is previously found in the literature. The main contributions here are:

- The formulation of a decentralized missile deflection problem as a transient stochastic game with communication. Ships are considered to be self-interested decision makers able to coordinate countermeasure deployment against a fixed number of missiles with known dynamics in real time.
- A theoretical foundation for Network Enabled Operations via the definition of NEOps equilibrium strategy, via a novel definition of correlated equilibrium in a stochastic game.
- A novel characterization of the correlated equilibrium as the solution to an optimization problem with bilinear constraints.
- An information theoretic analysis of the communication overhead required for implementation of the NEOps equilibrium strategy. Communication overhead is shown to be proportional to the mutual information between the random variables describing player actions in the correlated equilibrium strategy. The mutual information may also be used as a constraint in the optimization above.
- A comparison of communication architectures for implementation of the correlated equilibrium.

- Numerical examples illustrating the NEOps equilibrium strategy for a moderate-scale missile deflection games are presented, and the trade-off between communication and performance.

1.2.2 Contributions to Decentralized Sensor Activation Control

Decentralized operation is particularly important in wireless sensor networks, since central coordination of a large number of sensors is infeasible due to bandwidth and energy constraints. In Chapter 4, we present a game theoretic analysis for decentralized sensor activation control, based on energy and performance characteristics of the ZigBee communications protocol. The model allows individual sensors to trade-off their contribution to end-user awareness with the energy cost of activation. Besides the relative novelty of the model (existing game theoretic treatments in sensor networks tend to focus on activities such as communication and routing), the contributions of this chapter are as follows:

- A computationally efficient (see Section 4.2.3) decentralized, adaptive filtering algorithm is given for activation control. By adapting behaviour to a moving average performance history, the algorithm allows for tracking of the equilibrium as the environment changes (e.g. due to moving targets or sensor outage).
- Analysis of the satisfactory overall performance of the adaptive filtering rule, by showing converges to the *set* of correlated equilibria. Differential inclusions (rather than ordinary differential equations) are employed to analyze the convergence and tracking properties of the algorithm.
- An algorithm that exploits the CSMA structure of ZigBee/IEEE 802.15.4 for computing a maximum likelihood estimate of the number of nearby active sensors, to be taken as input by each adaptive filter. This is critical for estimating sensor interaction for the decentralized algorithm.
- A timer-based pricing mechanism for load-sharing between sensors, which prolongs effective network lifetime by reducing energy consumption. Fairness is enforced by forcing active sensors into a standby mode after a sufficient period.
- Implementation of the decentralized algorithms on a ZigBee test-bed in Matlab to demonstrate performance.

1.2.3 Contributions to Decentralized Dynamic Spectrum Access

Game theoretic analysis is particularly relevant when the decentralized system under consideration consists of components which are truly self-interested, in the sense that they respond to the motivations of independent, self-interested users. In Chapter 5, we consider such a scenario, with regards to decentralized dynamic spectrum access for a competitive system of cognitive radios. The game theoretic analysis considers cognitive radios which select a number of available channels for transmission, competing via carrier sense multiple access for bandwidth on each one in order to satisfy individual user demands. Besides implementing the decentralized, adaptive filtering algorithm of Chapter 4 in a novel, more complex setting to provide a mechanism for cooperative spectrum access between self-interested users, the chapter provides several contributions, including:

- Formulation of the decentralized channel access game for user rate satisfaction, given multiple shared channels whose availability varies in time due to the activity of primary, licensed users.
- Modeling of a simple CSMA protocol for channel sharing, including a specification of probability of channel access success, a maximum likelihood estimation procedure for determining channel contention based on channel access success history, and a specification of the probability of channel collisions.
- Formulation of a utility function which balances radio performance with a self-incurred social cost for depriving others of bandwidth, based on the quantities specified above and evaluated through the maximum likelihood estimation procedure. The utility function is motivated through analysis of its relationship with an intuitively reasonable measure of global optimality.
- Numerical analysis, including a study of how adjusting various parameters of the system affects overall performance, and a comparison with other game theoretic adaptive schemes.

1.2.4 Contributions to Self-Configuration of Sensors with Global Games

Sophisticated analysis of decentralized operation in sensor networks must account for the fact that individual sensors possess only imperfect information with which to make operational decisions. A game theoretic analysis can account for such situations by use of either Bayesian games [25], or, more tractably, with global games [17]. Chapter 6 represents, to the

best of our knowledge, the first application of global games to a decentralized engineering effort in a field such as wireless sensor networks. It models sensors making activation decisions as game players with *noisy* information about their environment, which predict the actions of nearby sensors and choose actions with imperfect information. The contributions in this chapter are as follows:

- Using the theory of global games, we motivate and establish conditions under which a simple threshold mode selection strategy provides competitively optimal (Nash equilibrium) behaviour for each sensor. This result holds even when each sensor receives its information in noise. Unlike previous work on global games, we consider the case where both observation noise and utility functions may vary between game players.
- Given a prior probability distribution for sensor information, which is observed in either Uniform or Gaussian noise, we formulate the Nash equilibrium mode selection thresholds as the solution to set of integral equations. The structure of the solution is examined for limiting low noise conditions, which are particularly important when sensors accumulate and average information over time.
- Equilibrium tests for the threshold strategy are given for both the Uniform and Gaussian noise cases. We also formulate a novel “near equilibrium” condition which provides a simple test for the threshold strategy under low noise conditions.
- A procedure for decentralized computation of the Nash equilibrium threshold mode selection strategy is also given, based on an iterative best strategy response between sensor types.

1.3 Review of Technology

Game theoretic design and analysis can potentially be applied to a wide variety of technologies, from the Internet (search, peer-to-peer systems, online auctions) to large-scale resource extraction and manufacturing technologies. In this thesis we apply the theory to automated systems enabled by wireless communications for defence, data gathering, and communication. For the reader’s convenience, these specific technologies are discussed below.

1.3.1 Missile Deflection and Network Centric Warfare

Missile deflection and defence technology rely on advanced research in guidance and control, target tracking, and Radar systems. Advanced missile deflection techniques rely on the

ability to use one’s own sensors (of which Radar is one) to determine the position and intentions of a threat, and then manipulate the missile’s sensors using physical or electromagnetic interference such as decoys or Radar jammers.

Threat Detection and Mitigation: The problem of threat detection and assessment using Radar systems is the subject of considerable research, employing a variety of disciplines such as optimal filtering, Bayesian networks, and context free grammars. A comprehensive overview of modern Radar tracking systems and threat assessment can be found in [14, 53], and due to the extensive research this thesis assumes a reliable system is available for use. Some advanced countermeasure techniques are outlined in [15, 20], but due to the military nature of these applications most detailed results are unavailable in the public domain. For this reason, this thesis takes a general approach to the missile deflection problem, using simplified models and missile dynamics to illustrate the underlying theory. Examples of similar approaches can be found in [18, 28, 39, 52].

Network Centric Warfare: One of the key motivations for a game theoretic treatment of missile deflection is the concept of Network Centric Warfare or Network Enabled Operations [4, 8]. This is not so much a technology as a mode of thinking about military operations as a problem in *information space*. In terms of missile deflection, this involves the integration of Radars and decision makers through real-time communications to maximize information and hence improve defensive capabilities. On the other hand, it prescribes that attacks should be focused not only on physical assets, but also on communication infrastructure for optimal results. This fact particularly motivates Chapters 2 and 3, which addresses how much communication infrastructure is necessary for operation of a missile deflection system, and what can be done if it is compromised.

1.3.2 Wireless Sensor Networks

Wireless sensor networks are a recent innovation stemming from advances in electronics miniaturization, environmental sensing technology, wireless communications, and advanced signal processing techniques [3, 26, 61]. The result is large, dispersed systems of inexpensive “sensor nodes” ideal for monitoring environments for which single-point-of-observation technologies fail. Wireless sensor networks have an advantage in that they can translate short-range signals such as ambient acoustic or magnetic activity into radio signals that can be efficiently propagated through multi-hop communication. In-network signal processing also increases efficiency by filtering and aggregating information *en route* to its destination.

Power Constraints: The flexibility of wireless sensor networks also provides its greatest challenges. Limited power supplies mean sensors must be extremely conservative in their activities if monitoring is to be done over long time periods. Since wireless communication is an energy-intensive activity, sensors must communicate in a highly efficient manner, only transmitting the most relevant information and using as little overhead as possible.

Challenges to Efficiency: There are several particular challenges faced in the quest for efficient operation of sensor networks. For example, collaborative target tracking with multiple sensors, energy-aware medium access control, routing, and querying, as well as sensor synchronization and tasking. Research activity in this area has been very high in recent years, with several conferences, books and journal issues being devoted entirely to the topic. Game theoretic approaches [35, 38, 40, 40, 49, 58, 59] are attracting considerable attention here and have displayed the capacity to provide flexible, effective solutions to many challenges in sensor networks. This thesis builds on these successes by taking a further, in-depth look at the game theoretic approach and extracting key design principles.

1.3.3 Cognitive Radio Dynamic Spectrum Access

Cognitive radio [32, 41, 42, 43] is an emerging technology born out of the potential for flexible, universal wireless communication proposed by software defined radio [19]. The general definition of cognitive radio has become quite broad, encompassing any communication features that can be learned by an intelligent computing system. The fundamental purpose of a cognitive radio remains, however, to provide sufficient “always on” communication in any environment, a feature that is addressed specifically by research into spectrum agility and dynamic spectrum access [2, 50].

There are, in fact, two approaches to the spectrum access problem in cognitive radio, which seeks to assign transmission strategies to cognitive radios so as to minimize interference with each other and with other licensed users. These are the spectrum underlay and spectrum overlay approaches. It is likely that these will eventually be combined into a flexible, working protocol for dynamic spectrum access that meets industry requirements.

Spectrum Underlay: The spectrum underlay approach involves using extremely low-power signals to avoid interference with neighbouring users, and is epitomized by ultra wideband technology [34, 57]. To ensure this, an *interference temperature* is defined for each user, and radios must estimate the extent to which their activity appears as noise to other users, and adjust to keep each others’ interference temperature below a threshold.

Spectrum Overlay: Spectrum overlay was introduced in [41] as “spectrum pooling” and investigated by the DARPA XG program [1] under the term “opportunistic spectrum access.” This approach involves opening up some radio frequencies to unlicensed access, under certain restrictions such as fairness and granting of precedence to licensed users. Several approaches have been advocated, including market-based allocation systems, priority assignments, and semi-regulated multiple access schemes.

Competitive Approach: The game theoretic approach to cognitive radio has a slightly different justification than for missile deflection and wireless sensor networks, as the radio environment is more competitive than cooperative. Each cognitive radio has its own goal: to obtain a required amount of bandwidth, and any overarching system goals are more or less irrelevant to this objective. While it is certainly possible to open up radio spectrum for unrestricted competition, radio spectrum is a scarce resource, and could be subject to unfair monopoly by resourceful users in this case. The careful placement of rules, along with associated penalties, is thus necessary for providing a practical cognitive radio environment, and the aim in this thesis is to appropriately mix cooperation and competition to produce a efficient, fair and flexible communication system.

1.4 Contents of this Thesis

The organization of this thesis is depicted in Figure 1.1, and discussed below. The thesis itself is written in a manuscript style as permitted by the Faculty of Graduate Studies at the University of British Columbia, with each chapter representing an independent research effort. Each chapter has its own motivation and literature review and can be read independently. For convenience all the references are placed in a single bibliography at the end of the thesis. The unifying theme is the use of game theory as an analytical tool to predict the behaviour of systems of autonomously operating system components. The main assumption is that each component can be accurately modelled as a rational player in a game, who maximizes their expected outcome as measured by a locally held utility function, as a response to expectations about the actions of other players.

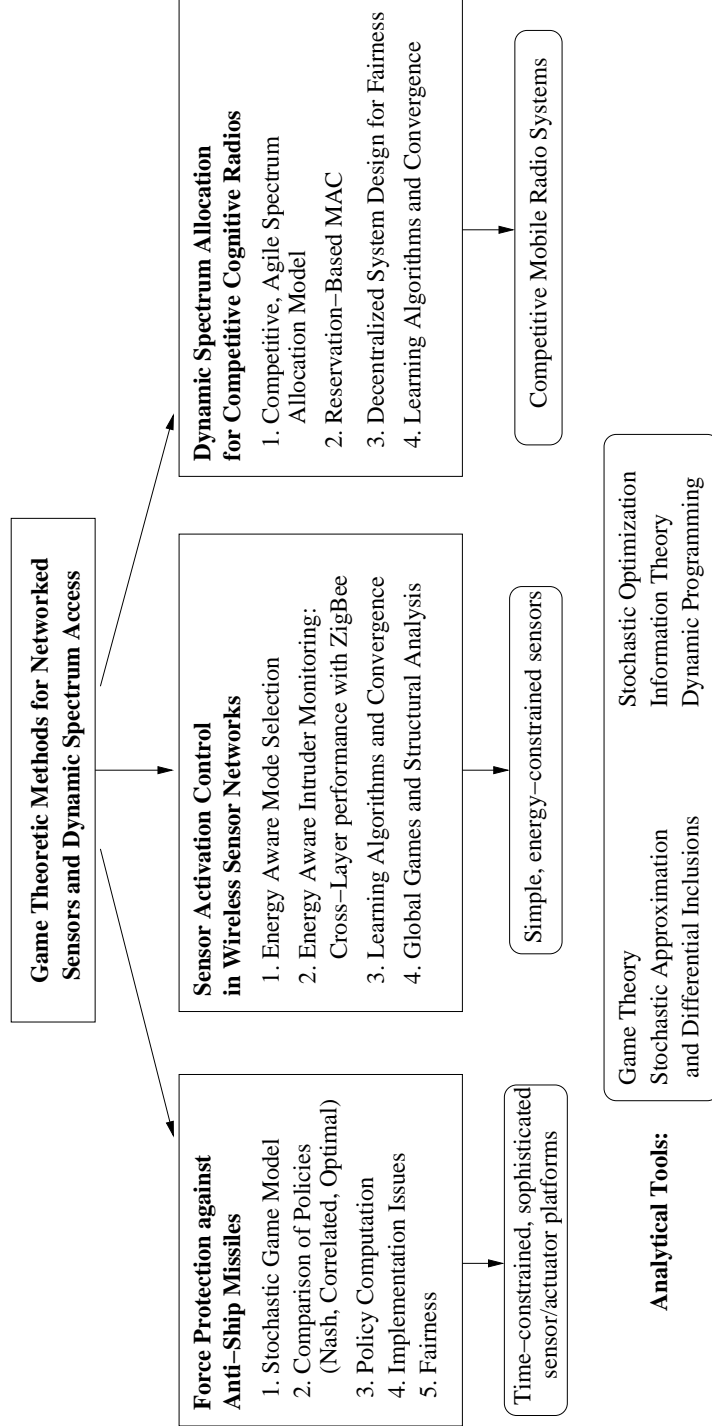


Figure 1.1: Organization of Thesis.

Networked Missile Deflection: In Chapter 2 a decentralized missile deflection model for Naval defence is analyzed as a stochastic game, with electronic countermeasure platforms filling the role of sensor-actuators. We give conditions for optimality of the equilibrium strategy when platforms are unable to communicate, and propose an optimization problem yielding the best available Nash equilibrium strategy when only offline communication and common observations are available. This Nash equilibrium solution is meant to be pre-computed by the platforms, and deployed through a lookup table at each platform to defend against known missile threats. We also discuss implementation issues, such as the problem of communicating situational awareness between platforms. Chapter 3 considers an extended scenario, in which platforms are capable of real-time coordination of effort, but may have differing objectives. In this case a correlated equilibrium best describes the optimal solution, and the improvement in using the correlated equilibrium is measured against the required communication overhead. In this case, the defensive solutions are deployed, for example, by a combination of pre-computed lookup tables; one for each ship, and optionally one for a coordinator which communicates with the task group. Practical schemes for implementation through real-time communication are discussed.

Wireless Sensors and Correlated Equilibrium: Chapter 4 applies a static game formulation to the problem of decentralized resource allocation in wireless, ad-hoc networks of energy-constrained sensors. We focus on the problem of activation control, by which sensors can conserve battery resources by entering a sleep mode when they are not needed for critical activities, such as target sensing. In this case, no pre-computed policy is given, but sensors learn an appropriate policy through repeated play. Specifically, each sensor trades off the expected marginal increase in global utility due to a report with the expected energy required to transmit data using the ZigBee protocol in a crowded environment. Since wireless sensors have limited computational resources, we start with a simple adaptive filtering protocol deployed at each sensor, and analyze its global performance in game theoretic terms. The result is that the sensor network is shown to learn and track the set of correlated equilibrium behaviour of the underlying game in a completely decentralized manner, and we give a rigorous convergence analysis, based on a stochastic approximation analysis.

Cognitive Radio and Correlated Equilibrium: Chapter 5 uses similar game theoretic techniques to solve a cooperative, decentralized dynamic spectrum allocation problem for a cognitive radio network. In this case, cognitive radios must select and share wireless communication channels when they are temporarily vacated by primary, licensed users.

Each radio's objective is to derive a channel allocation that satisfies its demand, while minimizing interference with others. A version of the adaptive learning scheme of Chapter 4 is proposed here for channel allocation. Estimation of the level of competition on each channel is critical to this learning scheme, and is also considered in detail.

Wireless Sensors and Global Games: Chapter 6 presents a novel game theoretic approach to the problem of self-configuration of sensors in a wireless sensor network. Decentralized operation and self-configuration in wireless sensor networks is investigated for energy-efficient area monitoring using the theory of global games. Given a large number of sensors which may operate in either an energy-efficient monitoring mode, or a costly tracking and data transmission mode, the problem of computing and executing a strategy for mode selection is formulated as a global game with diverse utilities and quality of observations. Each sensor measures a noisy version of environmental conditions, and determines whether to enter its "high-resolution" mode based on its expected contribution and energy cost. We use a game theoretic analysis to formulate Nash equilibrium conditions under which a simple threshold policy is competitively optimal for each sensor, and propose a scheme for decentralized threshold computation. The threshold level and its equilibrium properties depend on the prior probability distribution of environmental conditions and the observation noise, and we give conditions for equilibrium of the threshold policy as the noise varies.

Each chapter contains numerical simulations and analysis. Chapter 7 briefly summarizes the results of the thesis and provides concluding remarks.

Bibliography

- [1] DARPA: the next generation (XG) program. *www.darpa.mil/ato/programs/xg/index.htm*.
- [2] I. Akyildiz, W. Lee, M. Vuran, and S. Mohanty. Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Computer Networks*, 50:2127-2159, 2006.
- [3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks*, 38(4):393-422, 2002.
- [4] D. Alberts, J. Garstka, and F. Stein. *Network centric warfare: developing and leveraging information superiority*. CCRP Publications, second edition, 1999.
- [5] R. Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1:67-96, 1974.
- [6] R. Aumann. Correlated equilibrium as an expression of bayesian rationality. *Econometrica*, 55(1):1-18, 1987.
- [7] R.J. Aumann and S. Hart. Long cheap talk. *The Hebrew University of Jerusalem, Center for Rationality*, 2002.
- [8] S. Babcock. Canadian network enabled operations initiatives. *Proceedings of the 9th International Command and Control Research and Technology Symposium*, 2004.
- [9] I. Barany. Fair distribution protocols or how the players replace fortune. *Mathematics of Operations Research*, 17(2):327-340, 1992.
- [10] Elchanan Ben-Porath. Correlation without mediation: Expanding the set of equilibrium outcomes by "cheap" pre-play procedures. *Journal of Economic Theory*, 80(1):108-122, 1998.
- [11] M. Benaim, J. Hofbauer, and S. Sorin. Stochastic approximations and differential inclusions ii: Applications. Levine's bibliography, UCLA Department of Economics, May 2005.

- [12] J. Benyus. *Biomimicry: Innovation Inspired by Nature*. Perennial, 2002.
- [13] D. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1 and 2. Athena Scientific, second edition, 2001.
- [14] S. Blackman and R. Popoli. *Design and analysis of modern tracking systems*. Artech House, 1999.
- [15] L. Van Brunt, editor. *The glossary of electronic warfare*. EW Engineering, 1984.
- [16] A. Cahn. General procedures leading to correlated equilibria. *International Journal of Game Theory*, 33(1):21–40, 2004.
- [17] H. Carlsson and E. van Damme. Global games and equilibrium selection. *Econometrica*, 61(5):989–1018, 1993.
- [18] J. Cruz and M. Simann et. al. Game-theoretic modeling and control of a military air operation. *IEEE Transactions on Aerospace and Electronic Systems*, 37(4), 2001.
- [19] Dillinger, Madani, and Alonistioti. *Software defined radio : architectures, systems, and functions*. Wiley, 2003.
- [20] H. Eustace, editor. *The International Countermeasures Handbook*. EW Communications, sixth edition, 1980.
- [21] J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer-Verlag, 1997.
- [22] Francoise Forges. Universal mechanisms. *Econometrica*, 58(6):1341–64, 1990.
- [23] Francoise M Forges. An approach to communication equilibria. *Econometrica*, 54(6):1375–85, 1986.
- [24] D. Fudenberg and D. Levine. *The theory of learning in games*. MIT Press, 1999.
- [25] D. Fudenberg and J. Tirole. *Game theory*. MIT Press, 1991.
- [26] A.J. Goldsmith and S.B. Wicker. Design challenges for energy-constrained ad hoc wireless networks. *IEEE Wireless Comm.*, 9(4):8–27, 2002.
- [27] Olivier Gossner. Secure protocols or how communication generates correlation. *Journal of Economic Theory*, 83(1):69–89, 1998.

- [28] S. Gutman. On optimal guidance for homing missiles. *Journal of Guidance and Control*, 1979.
- [29] S. Hart and A. Mas-Colell, editors. *Cooperation: Game-Theoretic Approaches*, pages 199–217. Springer-Verlag, 1997.
- [30] S. Hart and A. Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.
- [31] S. Hart and A. Mas-Colell. *A reinforcement procedure leading to correlated equilibrium*, in *Economic Essays*, pages 181–200. Springer, 2001.
- [32] S. Haykin. Cognitive radio: Brain-empowered wireless communications. *IEEE Journal on Selected Areas in Communications*, 23(2):201–220, 2005.
- [33] P. Herings and R. Peeters. Stationary equilibria in stochastic games: structure, selection and computation. *METEOR Working Paper No. RM/00/31.*, 2000.
- [34] W. Hirt and D. Porcino. Ultra-wideband radio technology: potential and challenges ahead. *IEEE Communications Magazine*, 41(7):66–74, 2003.
- [35] R. Kannan, S. Sarangi, S. S. Iyengar, and Lydia Ray. Sensor-centric quality of routing in sensor networks. *IEEE Infocom*, 2003.
- [36] H.J. Kushner and G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer-Verlag, New York, NY, second edition, 2003.
- [37] Ehud Lehrer and Sylvain Sorin. One-shot public mediated talk. *Games and Economic Behavior*, 20(2):131–148, 1997.
- [38] Ning Li and Jennifer Hou. Localized fault-tolerant topology control in wireless ad hoc networks. *IEEE Transactions on Parallel and Distributed Computing*, 17(4):307–320, 2006.
- [39] Y. Lipman and J. Shinar. Mixed-strategy guidance in future ship defense. *Journal of Guidance, Control, and Dynamics*, 1996.
- [40] A.B. MacKenzie and S.B. Wicker. Game theory and the design of self-configuring, adaptive wireless networks. *IEEE Communications Magazine*, pages 126–131, November 2001.

- [41] J. Mitola. Cognitive radio for flexible mobile multimedia communications. *in: Proc. IEEE International Workshop on Mobile Multimedia Communications (MoMuC)*, pages 3–10, 1999.
- [42] J. Mitola. Cognitive radio: an integrated agent architecture for software defined radio. *Ph.D Thesis, KTH Royal Institute of Technology*, 2000.
- [43] J. Mitola. Cognitive radio for flexible mobile multimedia communications. *Mob. Netw. Appl.*, 6(5):435–441, 2001.
- [44] J. Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.
- [45] R. Nau, S. Gomez Canovas, and P. Hansen. On the geometry of nash equilibria and correlated equilibria. *International Journal of Game Theory*, 32(4):443–453, 2004.
- [46] M. Osborne and A. Rubenstein, editors. *A course in game theory*. MIT Press, 1994.
- [47] M. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, 1994.
- [48] J. Robinson. An iterative method of solving a game. *Annals of Mathematics*, 54:298–301, 1951.
- [49] C. Saraydar, N. Mandayam, and D. Goodman. Efficient power control via pricing in wireless data networks. *IEEE Transactions on Communications*, 50(2), 2002.
- [50] S. Shankar. Spectrum agile radios: utilization and sensing architecture,. *in: Proc. IEEE DySPAN*, pages 160–169, 2005.
- [51] L. Shapley. Stochastic games. *Proceedings of the National Academy of Sciences USA*, 39(10), 1953.
- [52] H. Sherali, Y. Lee, and D. Boyer. Scheduling target illuminators in naval battle-group anti-air warfare. *Naval Research Logistics*, 42, 1995.
- [53] M. Skolnik. *Introduction to radar systems*. McGraw-Hill, 2002.
- [54] V. Teague. Selecting correlated random actions. *Proceedings of Financial Cryptography*, 2004.
- [55] J. Vidal. *Fundamentals of multiagent systems: using netLogo models*. Unpublished, 2006. <http://www.multiagent.com/fmas>.

- [56] J. von Neumann and O. Morgenstern. *Theory of game and economic behavior*. Princeton University Press, 1944.
- [57] M. Win and R. Scholtz. Impulse radio: how it works. *IEEE Communications Letters*, 2(3):36–38, 1998.
- [58] Y. Xing and R. Chandramouli. Distributed discrete power control in wireless data networks using stochastic learning. *Proc. CISS, Princeton*, 2004.
- [59] Yiping Xing and R. Chandarmouli. Distributed discrete power control for bursty transmissions over wireless data networks. In *in Proc. CISS?* 2004.
- [60] T. Rabin Y. Dodis, S. Halevi. A cryptographic solution to a game theoretic problem. *Proceedings of CRYPTO*, 2000.
- [61] F. Zhao and L. Guibas. *Wireless sensor networks: an information processing approach*. Elsevier, 2004.

Chapter 2

Decentralized Force Protection Against Anti-Ship Missiles: Nash Equilibrium Approach ²

2.1 Introduction

Decentralized decision making in network centric warfare (NCW) is an area of research that seeks to balance integration and autonomy among military platforms. In [62], NCW is presented as a way to obtain and exploit an information advantage through shared awareness and communication; independent platforms share information and synchronize operations to achieve increased combat effectiveness. However, bandwidth limitations and the distributed nature of battlespace information often leaves platforms unable to fully integrate into a network centric force. This chapter considers such a scenario, in which platforms must make critical decisions independently, even while they work towards a common goal.

We focus on the problem of decentralized missile deflection in a Naval setting. This involves the rapid deployment of countermeasures (CM) on an extremely compressed time scale, so is an ideal candidate for decentralization. This is true even though missile deflection capabilities could be improved by coordinating CM-equipped ships, since coordination on such a rapid time scale may not be feasible, especially among a large and diverse fleet.

There are three important reasons to consider decentralized decision making in NCW. First, transmitting time-critical, high resolution battlespace data from all platforms to a central coordinator is bandwidth intensive, and the computational complexity involved in coordinating multiple ships is prohibitive. Second, decentralized architectures reduce vulnerability attacks in the information domain (jamming or spoofing), since isolated platforms are able to coordinate their actions even without communication. Finally, decentralization

²A version of this chapter has been accepted for publication. Maskery, M. O'Regan, C. and Krishnamurthy, V. (2007) Decentralized Algorithms for Netcentric Force Protection against Anti-ship Missiles, IEEE Transactions on Aerospace and Electronics Systems 43:4.

ensures robustness, since there is no reliance on communication equipment or on a central coordinator.

In this chapter we propose game-theoretic decision making algorithms for decentralized force protection against anti-ship missiles. We consider the following scenario. A task group, communicating over a common data network, faces an attack by a fixed number of missiles. The missiles appear randomly on the perimeter of a fixed physical space over time, and move toward the ships (targets) under known targeting and guidance laws. Each CM platform has hardkill and/or softkill weapons (countermeasures) that can be dynamically employed to deflect or destroy missiles in a statistically predictable way. Each platform observes missile states (e.g. bearing, range, etc.) and deploys countermeasures dynamically and independently, in light of the fact that other ships are doing the same. During an attack, weapons may act concurrently and must act in a complementary way. This scenario is formulated as a transient stochastic game, for which ships may compute a joint countermeasure (CM) policy that is in Nash equilibrium. This policy is executable in a decentralized fashion, thereby making it suitable for area defence.

The main contributions of this chapter are:

- A novel collaborative game theoretic formulation for force protection against anti-ship missiles.
- Formulation of a stochastic shortest path (transient stochastic) game and its Nash equilibrium solution.
- A Nash equilibrium solution based on state of the art algorithms for solving Markovian games.
- Simulation examples, with illustration of the sensitivity of the solution to changes in the missile dynamics.

The theory of Markov decision processes, [64, 81], and its extension to stochastic games, [72, 84] is used to formulate and solve the missile deflection problem. For a single platform, Markov decision theory provides the basis for computing optimal, dynamic missile deflection policies. These types of problems are well-studied, particularly in the Operations Research community. Of particular interest are [77, 78], which consider partially observed processes, and derive solutions by considering a fully observed approximate process. This suggests that our approach, which assumes the missile process is observed without error, is a useful one. Stochastic game theory, in turn, prescribes dynamic policies for multiple platforms, with optimality being replaced by the Nash equilibrium concept, which requires that each platform's policy is optimal given the policies of others.

There is a key difference between our approach and the more standard platform-centric approach considered in several papers in the literature. In the simplest instance of the platform centric model, one attacker and one defender play a pursuit-evasion game between defensive and anti-ship missiles, [66, 73, 76]. In this scenario, two missiles are viewed as players with clear targets, but no allowance is made for multiple defensive missiles to coordinate their actions. More generally, [70] models two forces (attacking and defending), engaging in a well-defined general sum stochastic game. Each force consists of multiple platforms, completely controlled by two respective commanders, with unlimited communication. However, since each force is considered a single unit from a decision making perspective, a decentralized model is again inapplicable. Finally, [85] considers a target illumination problem very similar to our multiple-ship, multiple-missile deflection scenario and computes an optimal scheduling solution for defeating the missile threat, which is implemented through full coordination. In contrast, we consider, here and in [79], a stochastic game with multiple platforms, with each platform being an independent decision maker, unable to fully communicate to its allies in real time. That is, ships play a game with *each other*, instead of with a missile. This perspective naturally allows us to consider decentralized solutions, which may be implemented when full coordination is impossible.

This chapter is organized as follows. In Section 2.2, we motivate the decentralized missile deflection problem and review the theory of stochastic shortest path games. In Section 2.3, we model the missile deflection problem as a stochastic game and describe the various dynamics and objectives of the game. In Section 2.4, we define the Nash equilibrium solution to the problem and propose an iterative solution method useful when all players have identical objectives. Section 2.5 considers the addition of CM constraints to the problem. In Section 2.6, two simplified examples are given to demonstrate the effectiveness of a Nash equilibrium policy. Finally, Section 2.7 summarizes the chapter. A glossary of variable definitions is provided in the appendix.

2.2 Motivation and Review of Stochastic Games

In this section, we motivate the decentralized missile deflection problem by presenting a decentralized architecture for CM deployment. We then briefly review the theory of stochastic games which will provide the framework for the decentralized scheme.

2.2.1 Decentralized Architecture for Implementing Countermeasures with a Common Air Picture

We begin with an overview of the decentralized architecture, in which CM platforms react independently to common observations of the missile threats. In our system, each CM platform uses local sensors to track missile threats, and shares this information with other platforms through a common database (the Track Cache). Using this common information, platforms deploy CM independently, using a local CM control system. The proposed scheme is outlined in Figure 2.1.

The use of local sensors to track missile threats allows for a significant reduction in communication. Platforms track the threat that they are currently engaging, and periodically store their information in a track cache, which provides common information to all agents. External track information, eg. from aircraft or sensor networks, is also reported to the track cache. A platform can switch threat engagements rapidly by looking up the required information in the track cache.

Communication is limited to track information, and is not present in decision making. This allows each platform to make time-critical decisions without having to wait for consensus or instruction from a centralized controller. The game theoretic approach allows coordination to emerge even though platforms act independently, since each platform attempts to predict and respond to the others' actions in an optimal fashion.

For the reader's convenience, we summarize the chapter in terms of Figure 2.1. The CM array contains the countermeasures described in Section 2.3.3, while the tracking systems are referred to in this section as well as Section 2.2.2. Missile dynamics will be discussed in Section 2.3.2. The main focus of this chapter, however, is on the CM control units. This includes the definition of the problem in game-theoretic terms (Section 2.3), the formulation of objectives (Section 2.3.4), and the computation of Nash equilibrium policies (Section 2.4 and Algorithm 2.4.1).

2.2.2 Motivation: Pulse Sequence Jamming over a Link 16 Network

We motivate our decentralized architecture through an example implementation. Assume a group of three ships implement a *centralized* CM system using a Link 16 network. Link 16 is a common tactical data link, used by some navy forces for the coordination of threat observations and action coordination. We assume a nominal bandwidth of 100 kbps for the data link. Each platform is targeted by a single missile, and records a threat radar pulse sequence from that missile. The CM mechanism operates by reflecting this pulse sequence

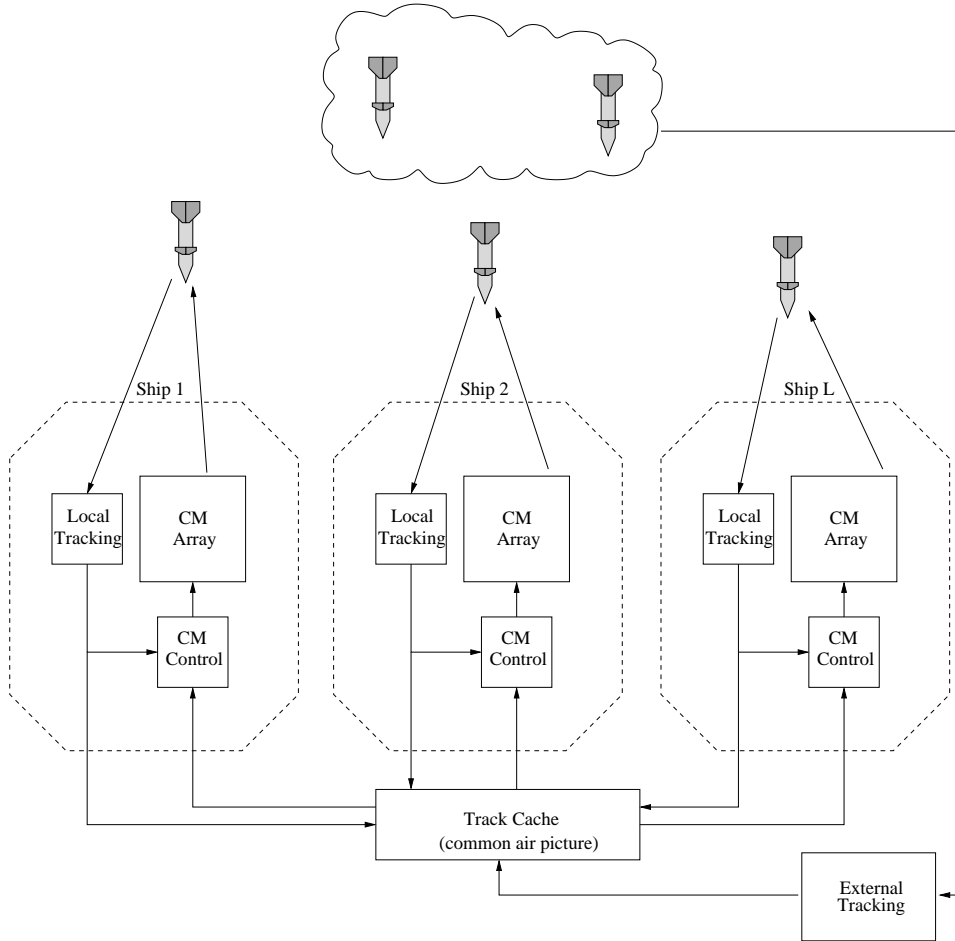


Figure 2.1: Overview of the decentralized missile deflection system. Agents merge local and external tracking information to ensure a common air picture, but otherwise make all countermeasure (CM) decisions locally.

after a small intentional delay, so as to confuse the missile targeting system. This strategy is based on a range gate pulloff CM strategy. The delay value is computed by a central CM system, which is transmitted to the ships. Assuming incoming missile threats travel toward the task group at a speed of 500 m/sec, we assume threat measurements and CM decisions are made every 0.1 seconds. We also assume that each platform has eight possible delay actions.

The bandwidth requirement for sending commands from a central CM decision system in the above scenario is about 30 bits/sec/user. This seemingly small data rate is significant, given that the Link 16 network may be supporting 1000-2000 processes of varying priority in an engagement scenario, but is nevertheless achievable using current and future networks. However, if communication between ships is reduced or eliminated due to fleet damage, jamming of the communication channel, or emission control limitations, or if the central CM control unit itself is damaged, the resulting loss of the centralized CM system will leave the ships isolated and unable to coordinate their CM actions to achieve maximum effectiveness. A decentralized architecture eliminates these vulnerabilities, making it much more robust in a realistic engagement setting.

2.2.3 Review of Stochastic Shortest Path Games

In this section we provide a brief overview of the components of a stochastic shortest path game. This combines the theory of stochastic games [72, 84], with the theory of transient Markov decision processes [64].

Definition of a Stochastic Shortest Path Game A stochastic shortest path game, derived from stochastic shortest path problems as in [64], is essentially a Markov decision process which reaches a terminal state in an almost surely finite number of steps. The objective is to choose an action policy which guides the game to the terminus with minimum cost (wherein cost is regarded as path length), when each player has independent input into the joint action at each step. An L -player stochastic shortest path game is a 4-tuple $(\mathbf{S}, \{\mathbf{U}^l, \mathbf{g}^l : l = 1, \dots, L\}, p)$ with the following interpretation. The state space \mathbf{S} is a finite collection of states, including a terminal state τ . The action set \mathbf{U}^l consists of the choices of player l . For each state $i \in \mathbf{S}$, define $\mathbf{U}^l(i) \subset \mathbf{U}^l$ to be the set of actions available. Denote the joint action set $\mathbf{U} = \mathbf{U}^1 \times \mathbf{U}^2 \times \dots \times \mathbf{U}^L$, with typical element $u = (u^1, u^2, \dots, u^L) = (u^l, u^{-l})$ for any $l \in L$. Player l 's one-period cost function is given by $\mathbf{g}^l : \mathbf{S} \times \mathbf{U} \rightarrow \mathbb{R}$. The transition probability $p : \mathbf{S} \times \mathbf{U} \rightarrow \Delta(\mathbf{S})$ specifies a probability distribution over next states, given the current state and actions, where $\Delta(\mathbf{S})$ denotes the space of all probability distributions

on \mathbf{S} . We write that $p(i, j; u)$ for the probability that the next state is j , given the current state and action are (i, u) , and require that $p(\tau, \tau; u) = 1$ for all u , and that $p(i, \tau; u) > 0$ for some reasonable pairs (i, u) in a sense to be made more precise below. We also require that the cost $g^l(\tau, u) = 0$ for all l, u .

Stationary Policies Each player in a game implements a policy for choosing actions in each state. As is standard in stochastic shortest path games, we restrict our attention to stationary randomized policies that are independently executable (which are known to be optimal for the problem at hand [63, 64]), as follows:

Definition 2.2.1 Let $\Delta(\mathbf{U}^l)$ represent the set of probability distributions over \mathbf{U}^l . An independently executable stationary randomized policy for platform l , denoted μ^l , is a function $\mu^l : \mathbf{S} \rightarrow \Delta(\mathbf{U}^l)$. For state i , the probability of deploying CM u is $\mu^l(i, u)$, and each distribution $\mu^l(i, :)$ has support at most $\mathbf{U}^l(i)$.

Since the cost and transition probabilities are a function of joint actions, we also define policy profiles as follows:

Definition 2.2.2 Given a collection of policies $\{\mu^1, \mu^2, \dots, \mu^L\}$. A policy profile μ is a function $\mu : \mathbf{S} \rightarrow (\Delta(\mathbf{U}^1) \times \Delta(\mathbf{U}^2) \times \dots \times \Delta(\mathbf{U}^L))$. For state i , the probability of deploying joint CM $u = (u^1, u^2, \dots, u^L)$ is

$$\mu(i, u) = \prod_{l \in L} \mu^l(i, u^l).$$

Also define μ^{-l} to be the policy profile of all agents but l , so that for any $\nu \in \Delta(\mathbf{U}^l)$, (ν, μ^{-l}) denotes a policy profile, with $\mu = (\mu^l, \mu^{-l})$.

Total Expected Costs Each policy profile μ induces expected one period costs and transition probabilities given by:

$$g^l(i, \mu) = \sum_{u \in U(i)} g^l(i, u) \mu(i, u), \text{ and,} \quad (2.1)$$

$$p(i, j; \mu) = \sum_{u \in U(i)} p(i, j; u) \mu(i, u). \quad (2.2)$$

Let $\mathbf{g}^l(\mu)$ denote a (column) vector of the $g^l(i, \mu)$ over all $i \neq \tau$, and let $P(\mu)$ denote a matrix of the $p(i, j, \mu)$ over all $i, j \neq \tau$. Let $\{X_n : n = 1, 2, \dots\}$ denote the evolution of a

transient Markov chain with transition matrix $P(\mu)$. For any $\mathcal{K} \subset \mathbf{S}$, the expected number of visits to \mathcal{K} given initial distribution $\alpha \in \Delta(\mathbf{S})$ is given by the *occupation measure* [63]:

$$f_{tc}(\alpha, \mu; \mathcal{K}) = \sum_{n=1}^{\infty} Pr(X_n \in \mathcal{K}) \quad (2.3)$$

The total expected cost to Player l under initial distribution α is defined as:

$$\mathbf{v}_{\alpha}^l(\mu) = \sum_{j \in \mathbf{S}} f_{tc}(\alpha, \mu; j) g^l(j, \mu). \quad (2.4)$$

For initial distribution δ_i with unit mass at $i \in \mathbf{S}$, we write:

$$\mathbf{v}^l(i, \mu) = \sum_{j \in \mathbf{S}} f_{tc}(\delta_i, \mu; j) g^l(j, \mu). \quad (2.5)$$

To ensure \mathbf{v}^l is finite, we restrict our attention to *proper policies* μ such that

$$\max_{i \in \mathbf{S}} (Pr(X_{|\mathbf{S}|-1} \neq \tau | X_0 = i, \mu)) < 1. \quad (2.6)$$

It follows that $P(\mu)$ is substochastic (at least one rowsum is less than one) and hence invertible. Define $\mathbf{v}^l(\mu)$ to be a column vector of the $\mathbf{v}^l(i, \mu)$ over $i \in \mathbf{S}$. This leads to the following theorem, which gives closed form expressions for the total expected cost under a policy μ :

Theorem 2.2.1 *The total expected costs under policy profile μ can be expressed as:*

$$\mathbf{v}^l(\mu) = (I - P(\mu))^{-1} \mathbf{g}^l(\mu), \text{ and}, \quad (2.7)$$

$$\mathbf{v}_{\alpha}^l(\mu) = \alpha(I - P(\mu))^{-1} \mathbf{g}^l(\mu), \quad (2.8)$$

Moreover, the total expected cost $\mathbf{v}^l(\mu)$ satisfies the recursion

$$\mathbf{v}^l(\mu) = \mathbf{g}^l(\mu) + P(\mu) \mathbf{v}^l(\mu). \quad (2.9)$$

Proof See Appendix.

Nash Equilibrium In the single player case, the theory of Markov decision processes yields an optimal policy that maximizes the total expected cost $\mathbf{v}(\mu)$. For multiple players, the equivalent condition is that the policy profile μ is a Nash equilibrium, defined as follows:

Definition 2.2.3 A policy profile μ is a Nash equilibrium if, for each $l \in L$, $i \in \mathbf{S}$ and every alternative ν^l ,

$$\mathbf{v}^l(i, \mu) \leq \mathbf{v}^l(i, (\nu^l, \mu^{-l})). \quad (2.10)$$

The Nash equilibrium profile is a collection of independently executable policies that are optimal from each player's perspective, and hence computable and implementable in a decentralized fashion. Thus, players may operate without explicit coordination, but their behaviour is still “competitively optimal” in the sense that each player's policy is an optimal response to the policies of the others.

In Section 2.3, we place the elements of the tuple $(\mathbf{S}, \{\mathbf{U}^l, \mathbf{g}^l : l = 1, \dots, L\}, p)$ into the context of the missile deflection problem. Computation of a Nash equilibrium policy profile is discussed in Section 2.4.

2.3 Decentralized Missile Deflection as a Stochastic Shortest Path Game

In this section we carefully construct the components of a stochastic game for the missile deflection problem. The game is described briefly as follows:

1. A multiple missile threat evolves in discrete time on the discrete state space \mathbf{S} , with the state at time $n = 0, 1, 2, \dots$ given by X_n .
2. A set of CM-equipped platforms L observe the threat process, and deploy countermeasures to protect the task group. At time n , platform $l \in L$ deploys countermeasure U_n^l from a given set $\mathbf{U}^l(X_n)$, *without knowledge of the CM actions chosen by others*. This results in a joint countermeasure $U_n = (U_n^1, U_n^2, \dots, U_n^L)$, which belongs to an appropriate product space $\mathbf{U}(X_n)$.
3. Each platform $l \in L$ associates a cost $g^l(i, u)$ with each non-terminal state $i \in \mathbf{S}$ and action $u \in \mathbf{U}(i)$, and seeks a competitively optimal policy μ^l in the form of a Nash equilibrium.
4. The missile threat reacts to the joint countermeasure, defining a stochastic process $\{X_n, n = 0, 1, 2, \dots\}$ on \mathbf{S} , such that

$$Pr(X_{n+1} = j | X_n = i, U_n = u) = p(i, j; u). \quad (2.11)$$

These probabilities are assumed known for every $i, j \in \mathbf{S}$ and $u \in \mathbf{U}(i)$.

We assume that the missile threat process reaches a terminal state τ in (almost surely) finite time, which corresponds to a termination of the game either through missile hits or a successful evasions, or a combination of both. The key assumptions of our model are as follows:

- The theater of operations is modeled as a discrete space. Since a platform obtains its awareness through digital sensors which necessarily quantize threat information, this assumption is, in a sense, natural. Moreover, if sensor measurements are noisy, a coarser discrete state space serves (heuristically) to reduce this noise by “rounding” measurements to their nearest discrete state.
- Missiles move in discrete time. This reflects the fact that known sensor systems return threat observations only at discrete, periodic intervals.
- The task group is considered to be stationary, due to the relative speed difference between missiles and ships, the short engagement time, and the fact that ship maneuvers may be restricted.
- Sensors are assumed to yield accurate information about the missile position and velocity. Considering the quality of service of sophisticated radar tracking systems, and the fact that tracking often occurs on a faster time scale than decision making, thus averaging out noise, this is reasonable.
- Missiles do not coordinate their actions. This is a reasonable simplifying assumption given the scope of this chapter, as it allows us to avoid complications in the missile dynamics. It is quite possible that missiles will coordinate their actions in reality, in which case the techniques in this chapter still apply, but the dynamics become more complex.

When sensor measurements are subject to noise, the scenario presented here provides us with an upper performance bound. In addition, one can use the results here to construct a certainty equivalent (suboptimal) controller for partially observed problems [64]. We also refer the reader to the sensitivity analysis of Section 2.6.2, which suggests that measurement errors are not necessarily detrimental to performance.

In the remainder of this section, we describe the components $(\mathbf{S}, \mathbf{U}^l, \mathbf{g}^l, p)$ of the stochastic game in detail.

2.3.1 State Space of the Multiple Missile Threat

We begin by describing the state space \mathbf{S} of the stochastic missile deflection game. The state space of a single missile m is described first, which is then extended to the multiple missile threat. At time n , missile m has state given by:

$$\begin{pmatrix} Y_n^m \\ Z_n^m \end{pmatrix} = \begin{pmatrix} \text{Position of missile } m \text{ at time } n \\ \text{Target of missile } m \text{ at time } n \end{pmatrix}. \quad (2.12)$$

This state is an element of the space:

$$\mathbf{S}^m = \{B \times T, r, \tau\}. \quad (2.13)$$

Here, either $Y_n^m \in B$ and $Z_n^m \in T$ or the missile is in one of two special states r, τ . The battlespace B represents the physical missile location, while T represents its target. Since the missile target is also a location, we take $T \subset B$. We also define $G \subset T$ to be the set of true target positions occupied by the task group. Finally, the pre-engagement state r corresponds to missile m being a potential but as-yet untracked threat, while the terminal state τ , corresponding to a missile has reached a target, terminated its flight, or has acquired a trajectory that ensures it is no longer a threat. For states $i \in B \times T$, we denote i by (i_y, i_z) , where i_y and i_z denote position and target components, respectively.

The space B is taken to be a two-dimensional grid, sufficient for sea-skimming missiles. A 600×600 grid can describe the position of a missile over a 30 km range with 50 m precision, suitable for speeds of 500 meters per second and threat measurements made 10 times per second.

To extend the threat to multiple missiles, let M denote the number missile threats in a given scenario, including potential missiles (in state r). For ease of notation, M also denotes the set of missiles. The threat state at time n is defined as

$$X_n = (Y_n^1, Z_n^1, Y_n^2, Z_n^2, \dots, Y_n^M, Z_n^M). \quad (2.14)$$

This is simply an element of product state space \mathbf{S} , defined as

$$\mathbf{S} = \mathbf{S}^1 \times \mathbf{S}^2 \times \dots \times \mathbf{S}^M. \quad (2.15)$$

For simplicity, we label state $(\tau, \tau, \dots, \tau) \in \mathbf{S}$ as simply τ , corresponding to the termination of all missile threats. A generic state $i \in \mathbf{S}$ is written as $(i_y^1, i_z^1, i_y^2, i_z^2, \dots, i_y^M, i_z^M)$, or simply

as (i^1, i^2, \dots, i^M) when positions and targets do not need to be distinguished.

2.3.2 Dynamics of a Radar-Guided Multiple Missile Threat

The dynamics of the missile process, described by the transition probability function p is described in this section. We assume that each missile evolves independently, so the state (Y_n^m, Z_n^m) as in (2.12) evolves according to a Markov process for each missile m , controlled in part by CM actions. The transition probabilities specify the dynamics of the process, which can be separated into two parts: the physical dynamics and the targeting dynamics. These are defined as follows:

Definition 2.3.1 *For each joint CM action U_n at time n , the physical and targeting transition probabilities of missile m are separately given by*

$$\begin{aligned} p_y^m(i, j; z, u) &= \Pr(Y_{n+1}^m = j | Y_n^m = i, Z_n^m = z, u_n = u), \\ p_z^m(i, j; y, u) &= \Pr(Z_{n+1}^m = j | Z_n^m = i, Y_n^m = y, u_n = u). \end{aligned} \quad (2.16)$$

These probabilities are assumed to be conditionally independent, hence the overall transition probabilities are given by

$$p^m(i, j; u) = \Pr((Y_{n+1}^m, Z_{n+1}^m) = (j_y, j_z) | (Y_n^m, Z_n^m) = (i_y, i_z), u_n = u) \quad (2.17)$$

$$= p_x^m(i_y, j_y; i_z, u) p_z^m(i_z, j_z; i_z, u). \quad (2.18)$$

The transition probabilities are assumed given, and the following natural conditions can be imposed on them, for all controls u , and all active states i, j :

$$\begin{aligned} p^m(i, \tau; u) &= 1 \text{ whenever } i_x \in G, \text{ (missile terminates upon reaching a target)} \\ p^m(\tau, \tau; u) &= 1, \text{ (terminated missiles do not recover)} \\ p^m(r, j; u) &= \beta^m(j), \text{ (activation probabilities are independent of control).} \end{aligned} \quad (2.19)$$

β^m is the transition probability distribution out of the pre-engagement state r . For example, $1 - \beta^m(r)$ is the probability that a potential missile will become active at the next time index. Three example probabilistic models for the missile processes are given below.

The proper policy assumption (2.6) requires that the missile may reach state τ from any state with positive probability. This is not too restrictive a condition, and we assume it holds here. It simply states that the missile threat cannot carry on forever without terminating.

Example *Probabilistic model for the physical guidance system.* Assume the missile operates according to proportional guidance [65]; the missile moves on an intercept course towards a target by predicting its position at the time of impact. However, because the missile may only make noisy position and velocity estimates of its target, occasional deviations from the proper course may occur. Suppose a missile at a is on a straight-line intercept course that should take it next to point $b(a, z)$ on its way to target position z . A simple model for introducing random guidance errors is to specify:

$$\begin{aligned} p_y^m(a, b(a, z); z, u) &= 1 - q \\ p_y^m(a, b^+(a, z); z, u) &= q/2 \\ p_y^m(a, b^-(a, z); z, u) &= q/2 \end{aligned} \tag{2.20}$$

for a given z, u and some small error probability q , where b^+ and b^- are points in B to the immediate left and right of b .

Example *Probabilistic model for the targeting system.* The targeting transition probabilities depend on some programmed decision rules of the missile, but targets can also “drift” if the missile is viewed as a receiver operating in a noisy environment. Suppose the missile has chosen a target at location z , and the radar receives information about the target position over a discrete memoryless channel [69]. The channel randomly reports the position as z' when in fact it is z with probability $p_z^m(z, z'; y, u)$ and the missile updates z accordingly. The distortion may or may not vary with the missile position. Effective countermeasures may be modeled as systems in which $p_z^m(z, z'; y, u)$ decreases with u .

Example *Probabilistic model for missile decoys*

It is conceivable that some of the missiles in M may be decoys, intended to reduce the effectiveness of CM by forcing agents to spread their resources too thinly. If decoys cannot be discriminated, then agents will be forced to treat every missile as real. However, if decoys are detectable, then agents may remove them from consideration as they are detected, thereby increasing the effectiveness of the defence system.

Assume that the dynamics of the missile deflection problem are given by the transition probabilities $p^m(i, j; u)$ for all $m \in M$, $i, j \in \mathbf{S}^m$, and $u \in U(i)$, and assume that all agents independently discover missile m to be a decoy with probability

$$Pr(\text{missile } m \text{ is detected to be a decoy in state } i | \text{missile } m \text{ is a decoy}) = f(i). \tag{2.21}$$

It is assumed that no false detections occur. If missile m has an *a priori* probability of

being a decoy given by

$$Pr(\text{missile } m \text{ is a decoy}) = g(m). \quad (2.22)$$

Then, we have the following proposition:

Proposition 2.3.1 *Given that decoy detections are independent, and assuming that false decoy detections are negligible, the dynamics of the missile deflection problem can be replaced with:*

$$\begin{aligned} p^m(i, j; u) &:= p^m(i, j; u) (1 - g(m)f(i)), \quad j \neq \tau, \text{ and,} \\ p^m(i, \tau; u) &:= 1 - (1 - p^m(i, \tau; u) (1 - g(m)f(i))). \end{aligned} \quad (2.23)$$

Proof If a missile is revealed to be a decoy, it ceases to be a threat, and hence jumps to state τ . Since no false detections occur, we have

$$Pr(\text{missile } m \text{ is detected to be a decoy in state } i) = g(m)f(i). \quad (2.24)$$

The missile transitions to j only if it is not revealed to be a decoy. Hence the transition probabilities are discounted to

$$Pr(Y_{t+1}^m = j | Y_t^m = i) = p^m(i, j; u)(1 - g(m)f(i)). \quad (2.25)$$

The second part of (2.23) is then derived from the requirement that probabilities sum to one:

$$1 - \sum_{j \neq \tau} [p^m(i, j; u)(1 - g(m)f(i))] = 1 - (1 - p^m(i, \tau; u) (1 - g(m)f(i))). \quad (2.26)$$

■

Once decoy detection is incorporated into the model via the modified transition probabilities (2.23), agents will automatically account for the possibility of decoys in their optimal policies. In essence, agents will adjust their optimal policies to slightly favor states in which detection probability is high, in hopes of driving some missiles to termination.

We assume that each missile evolves independently (they do not communicate). In this case the transition probabilities of the threat process may be written as

$$p(i, j; u) = \prod_{m=1}^M p^m(i^m, j^m; u). \quad (2.27)$$

This formulation allows us to consider a variety of threat situations. Not only does each distinct value of M represent a fundamentally different threat situation, but the potential for mixtures of different missile types leads to a combinatorially large number of situations for each M . In practice, it is recommended that only a small number of threat situations be considered, and that these are chosen to be representative of the types of situations anticipated.

2.3.3 Countermeasure Actions

The action set \mathbf{U}^l of the stochastic game will be described in this section. The action set of each ship depends on the type and number of CM capabilities available. We assume that the CM actions available for any threat state are enumerated from one to $|U^l(i)|$. The effects of each CM action are specified by the transition probabilities p as in Section 2.3.2. In this section, we consider the CM actions in a qualitative way.

There are a wide range of countermeasures (hardkill and softkill weapons) for deflecting active radar homing anti-ship missiles. Softkill weapons target the missile radar, deceiving or disorienting the missile to so as to counteract its targeting system. Chaff, offboard active or passive decoys, and onboard active jammers are some of the examples considered, the details of which can be found in [68, 71, 80]. Hardkill weapons are directed to intercept the missile and actively destroy (kill) it through direct impact or explosive detonation in the proximity of the threat missile. In this section we briefly outline how each of these can be modeled, in a general way, as a CM choice affecting the transition probabilities of the missile process.

Passive Decoys and Chaff Chaff consists of thin metallic strips launched into the air, which supply false target echoes to the missile radar. Similarly, discrete passive decoys are lightweight structures that float and have a high radar cross section. The decoys are designed with a number of corner reflectors which mimic target radar signatures. In this model, chaff is restricted in the sense that once a ship launches a chaff cloud in a desired location, no further control choices can be made, although the chaff may drift with the wind. This is a CM constraint, which may be incorporated into the model by the methods of Section 2.5.3. Another model would allow a series of chaff launches while the ship moves away from the resulting blooms. Chaff bloom points are range limited. If a chaff or passive decoy launch is part of the current CM joint action, the transition probabilities $p(i, j; u)$ are affected by increasing the probability that missiles will target the chaff cloud or decoy.

Off-board Active Decoys and Onboard Active Jammers These countermeasures also affect the missile transition probabilities by providing false targets, but can be dynamically controlled. Off-board active decoys may be placed on moving platforms and contain radio frequency repeaters to radiate a ship-like radar signal. The decoy is directed by control of the platform motion. The decoy moves slowly away from the vessel hopefully drawing the attacking anti-ship missile away from its intended target at the same time. The limits on velocity can be modeled by the control constraints discussed in Section 2.5.3. Active onboard jammers emit electromagnetic radiation designed to confuse a missile radar, so that its target position is lost or effectively moved. For example, cross-polarized and cross-eye jamming cause angle errors. Range gate pulloff acts similarly but may be modeled as causing only range errors instead of angle errors.

Hardkill Measures Hard kill measures, such as anti-missile rockets and close-in weapon systems, affect a threat missile by driving the missile directly to termination. To model a hard kill measure against missile m , the transition probability $p^m(i, \tau; u)$ is set roughly equal to the probability of success of the hardkill measure, in a similar manner to (2.23). To reduce the number of possible control choices, hard kill measures can be assumed to be directed at the most immediate threat.

2.3.4 Platform Objectives and State Costs

In this section we formulate two defensive objectives for deploying countermeasures, and associate these with the cost function \mathbf{g}^l of the stochastic game. Two special categories of objectives are positional objectives, in which $g^l(i, u)$ is independent of u , and collaborative objectives, in which $g^l(i, u)$ is identical for all l . For simplicity, we consider only positional objectives, and later we present an algorithm designed for collaborative ones. Collaborative objectives are especially attractive, since agents will not be tempted to save themselves by endangering others. Positional objectives are sufficient when the goal of defending targets outweighs considerations of the types of CM used.

For active missiles, we define the following:

Definition 2.3.2 For $l \in L$, define $G(l)$ to be the set of targets defended by platform l . We assume all targets are protected by at least one platform, so that

$$G = \bigcup_{l \in L} G(l). \quad (2.28)$$

The target sets give us a context in which to consider the defensive objectives below.

Minimizing the Probability of Hit

A reasonable defensive goal for platform l is to minimize the probability that the missiles will hit its target set $G(l)$ from any initial state. Consider the cost function

$$g^l(i) = I(i_y^m \in G(l), \text{ some } m), \quad (2.29)$$

where I is the indicator function.

Minimizing (2.29) results in a policy that minimizes the probability of hit. This is established by the following proposition:

Proposition 2.3.2 *If \mathbf{g}^l is as in (2.29), then the i^{th} entry of $(I - P(\mu))^{-1}\mathbf{g}^l$ is exactly the probability of reaching $G(l)$ from state i under joint policy $\mu = (\mu^1, \mu^2, \dots, \mu^L)$.*

Proof As in Theorem 2.2.1, the ij^{th} entry of $(I - P(\mu))^{-1}$ gives the expected number of visits to state j starting from state i and following joint policy μ . By (2.19), states j such that $g^l(j) = 1$ are visited at most once, so the expected number of visits to such states is also the probability that they will ever be reached. The matrix multiplication $(I - P(\mu))^{-1}\mathbf{g}^l$ sums the probabilities, which are mutually exclusive events, hence the result is simply the probability of reaching $G(l)$ under policy μ . ■

The quantity $(I - P(\mu))^{-1}\mathbf{g}^l$ can be obtained by solving the equation

$$q_\mu^l = P(\mu)q_\mu^l + \mathbf{g}^l \quad (2.30)$$

for q_μ^l . If the $G(l)$ are disjoint, summing the probabilities of hit over all agents will give the overall hit probability (the probability that the target set G is reached). Thus, a measure of global performance can be obtained even when the platform objectives are not collaborative, for example as the solution q_μ to

$$q_\mu = P(\mu)q_\mu + \sum_{l \in L} \mathbf{g}^l. \quad (2.31)$$

Multiplication of q_μ by initial distribution α gives a real-valued performance measure for a given policy.

Maximizing the Minimum Distance

Another objective is to maximize the minimum distance achieved between missiles and targets over the entire history of the process. Under this objective, agents endeavor to keep each missile at least as far away from its target set as it has been in the past. This is not immediately a Markovian objective; if a missile threatens to enter some state j , platform l will react differently depending on the threat history. If missiles have previously come closer than j , then the platform will not expend as much effort repelling this particular threat, since it has survived worse threats and likely has more immediate ones.

The practice of reacting differently depending on the threat history requires some explanation. A policy which seeks to prevent missiles from coming closer than they ever have in the past is prone to defend ships less aggressively than a policy which simply seeks to prevent missiles from coming closer than they currently are. (This second policy is easily implementable without the state augmentation of this section, simply by penalizing a player whenever a missile moves closer than it was in the previous state.) However, a policy which depends on past as well as present states opens up rich possibilities for stochastic games. For example, as a missile attack progresses, the primary objective may not be to keep missiles as far away as possible, but to maintain a previously established safe perimeter, in which maneuvers can be performed. The practice of maximizing the minimum distance can serve to spread defensive efforts among different missile threats, providing a more robust perimeter at the possible cost of a temporary loss of space.

A further justification for maximizing the minimum distance is that there may be a perimeter at which countermeasures are particularly effective, however this perimeter may be relatively close to the task group. By relaxing the objective to account for historical distances, the task group can "draw in" missiles to this effective range before engaging aggressive countermeasures, which may be a more strategic use of countermeasures than the alternative, non-history dependent policy.

To account for historical distances, we must augment the state space of the problem. For $i \in \mathbf{S}$ and all sets $A \subset B$, where B is the battlespace as in (2.13), define the distance from i to A by $d(i, A)$. A particularly convenient distance function is defined as follows.

Definition 2.3.3 Fix $\varepsilon > 0$. Partition \mathbf{S} into $\{D_0^l, D_1^l, \dots\}$, with $i \in D_k^l$ if and only if $G(l)$ can be reached from i in k steps with some minimal probability ε^k . Define $d(i, G(l))$ to be the minimum number of transitions it takes to reach $G(l)$ from i with this minimal

probability. That is,

$$\begin{aligned} d(i, G(l)) &= k : i \in D_k^l \\ &= \min_{\mu} \left(k : \Pr(X_k \in G(l) | X_0 = i, \mu) > \varepsilon^k \right). \end{aligned} \quad (2.32)$$

Note that the partition can be defined recursively, with $D_0^l = \{i \in \mathbf{S} : i_y^m \in G(l), \text{ for some } m\}$, and

$$D_k^l = \left\{ i \in \mathbf{S} : p_x^m(i_y^m, j_y^m; z, u) > \varepsilon, \text{ for some } m, z, u, \text{ some } j_x^m \in D_{k-1}^l \right\}. \quad (2.33)$$

If $d(i, G(l)) = \infty$, set $d(i, G(l)) = 0$ instead. Let $\mathcal{D}^l = \{0, 1, 2, \dots, K^l\}$ represent the possible values that $d(i, G(l))$ can take on.

Remark From the proper policy assumption (2.6) in Section 2.2.3, it is always possible reach τ in at most $|\mathbf{S}|$ steps. It follows that the maximum distance cannot exceed $|\mathbf{S}|$, since the case $d(i, G(l)) = \infty$ is excluded. There is no harm in this exclusion, since Platform l is safe whenever $d(i, G(l)) = \infty$, so the policy here does not matter.

Remark The distance $d(i, G(l))$ measures the minimum number of steps it would take for one of the missiles in state i to reach either the set $G(l)$ or the terminal state. It depends on the transition probabilities of the missile process, but not on the particular CM sequence used. The distance function defined in this way also has the attractive property of being integer valued.

Define a new stochastic process on the state space

$$\mathbf{S} := \mathbf{S} \times \mathcal{D}^1 \times \dots \times \mathcal{D}^L. \quad (2.34)$$

We define a new process to track both the missile state X_n and the previously achieved minimum distance to each target set $G(l)$. Writing $i_{\mathcal{D}^l}$ as the component of i on \mathcal{D}^l , the new process evolves such that

$$j_{\mathcal{D}^l} = \min(d(i, G(l)), i_{\mathcal{D}^l}) \quad (2.35)$$

for all i, j such that $p(i, j; u) > 0$.

To efficiently formulate the new process, we group agents with the same objective together to form a reduced state space. That is, we define \bar{L} by removing from L any platform

l if $G(l) = G(m)$ for some $m < l$. If we denote the original state space by $\bar{\mathbf{S}}$, the corresponding states by \bar{i} , and the original transition probabilities in (2.27) by $\bar{p}(\bar{i}, \bar{j}; u)$, then the transition probabilities on the new state space \mathbf{S} reflecting the history of minimum distances are given by

$$p(i, j; u) = \bar{p}(\bar{i}, \bar{j}; u) \prod_{l \in \bar{L}} I[j_{\mathcal{D}^l} = \min(d(i, G(l)), i_{\mathcal{D}^l})], \quad (2.36)$$

$i_{\mathcal{D}^l}$ is interpreted as the historic minimum distance achieved by the process before it reached state i . $d(i, G(l))$ is the current distance of the process in state i , so the indicator functions in (2.36) simply update the historic minimum distance to target set l . The new stochastic process is non-increasing on each new dimension D^l .

To maximize the minimum distance, the costs are defined so that a platform is penalized whenever the missile reaches a historic new minimum distance. A platform will then compute a policy which minimizes these penalties, and hence maximizes the minimum distance. Specifically, we set

$$g^l(i, u) = \begin{cases} \max(0, i_{\mathcal{D}^l} - d(i, G(l))), & i \neq \tau, \\ 0, & i = \tau. \end{cases} \quad (2.37)$$

Since the number of states in \mathbf{S} increases exponentially with both the number of target sets and the number of missiles, the objective of maximizing the minimum distance is most useful in a purely collaborative game setting, so that (2.34) becomes simply $\mathbf{S} = \mathbf{S} \times \mathcal{D}$.

Remark The distance concept suggests a useful heuristic for constructing candidate optimal policies under multiple missile threats. If an platform is able to resolve the distance from each individual missile to its target set, then a likely good decision is to concentrate countermeasures first on the missile that will be first to arrive. This results in a significant reduction in problem complexity, without significantly degrading performance.

As a final note, we mention that since the extra dimensions are artificial, the distance function can be coarsened, refined, or changed without losing information on the missile dynamics, although the optimal policies will in general be changed. The resulting trade off between complexity and performance is a model design decision.

Priority and Sacrifice

In some situations, it may be desirable to draw missiles toward a less valuable or more robust asset (e.g. frigate), so as to protect a more valuable asset (e.g. aircraft carrier).

One way to assign value to assets is to include it in the target sets $G(l)$ of more or fewer platforms, so that higher valued assets are protected by more platforms. Another way is to assume each platform l assigns value $v^l(i)$ to each asset location $i \in G(l)$ and use the objective of Section 2.3.4. Agent l can then use the weighted cost function

$$g^l(i) = v^l(i)I(i_y^m \in G(l), \text{ some } m) \quad (2.38)$$

to produce a policy that defends assets differently in proportion to their value. Hence some assets will be readily sacrificed for the benefit of others. The situation is more complex when the objective of Section 2.3.4 is considered. In this case, assets may be discarded from the target set $G(l)$ as the minimum distance decreases. However, the location of assets within the task force will then play an important role in how ships are sacrificed.

Regardless of the method used, it should be emphasized that task groups will almost always have assets of differing priorities. Sacrificing some assets to defend others is a valuable tactic in task force protection, and is the inevitable outcome of any priority assignment.

2.4 Computation of Decentralized Missile Deflection Policies

In this section we consider the decentralized computation and execution of a joint CM policy. In particular, we consider policies that are optimal responses to each other, and hence in Nash equilibrium. In this case, the joint policy will also be independently executable, so agents need not communicate during a missile attack. Section 2.4.3 presents the main algorithm for computing Nash equilibrium policies, while Sections 2.4.1 and 2.4.2 describe the efficient linear programming techniques used to carry out the main algorithm.

2.4.1 Computation of an Optimal Response for a Single Ship

Since a Nash equilibrium is a collection of optimal responses, we first consider the single-platform optimization problem. If μ^{-l} is held fixed, platform l can compute an optimal stationary policy μ^{*l} by viewing the problem as a stochastic shortest path Markov decision process. This will specify which CM actions should be used for a given threat state, in order to minimize the total expected cost to the platform. Such problems have been extensively studied, for example in [64, 72, 81], and dynamic programming-based solution methods are well known.

We have already discussed stationary policies and total expected costs in Section 2.3. In this section we summarize known results for computing optimal CM policies in terms of stochastic shortest path problems. We focus on the linear programming solution, but many other methods, such as value iteration, policy iteration, Monte Carlo methods and approximation can also be used to obtain suitable policies. The optimal response can be obtained via the following proposition.

Proposition 2.4.1 *Let α be an initial probability distribution on \mathbf{S} , such that $\alpha(i) > 0$ for all $i \neq \tau$, and fix μ^{-l} . The optimal response of l , μ^{*l} , has costs $\mathbf{v}^l(i, (\mu^{*l}, \mu^{-l}))$ given by the solution to*

$$\begin{aligned}
 & \text{maximize } \sum_{i \neq \tau} \alpha(i) J^l(i) \\
 & \text{subject to } (\forall u \in U^l(i), i \neq \tau) \\
 & J^l(i) \leq g^l(i, (u, \mu^{-l})) + \sum_{j \neq \tau} p(i, j; (u, \mu^{-l})) J^l(j).
 \end{aligned} \tag{2.39}$$

Furthermore, μ^{*l} can be obtained by solving the dual linear program for π^l :

$$\begin{aligned}
 & \text{minimize } \sum_{i \neq \tau, u \in U^l(i)} g^l(i, (u, \mu^{-l})) \pi^l(i, u) \\
 & \text{subject to } (\forall j \in \mathbf{S}) \\
 & \sum_{u \in U^l(j)} \pi^l(j, u) = \sum_{i \neq \tau, u \in U^l(j)} p(i, j; (u, \mu^{-l})) \pi^l(i, u) + \alpha(j), \\
 & \pi^l(j, u) \geq 0 \quad \forall u \in U^l(j).
 \end{aligned} \tag{2.40}$$

Setting

$$\mu^{*l}(i, u_r) = \frac{\pi^{*l}(i, u_r)}{\sum_{u \in U^l(i)} \pi^{*l}(i, u)}, \tag{2.41}$$

where π^{*l} solves (2.40), yields the optimal policy.

It is important that the linear programs (2.39) and (2.40) are duals of each other, [64, 72, 81]; they essentially solve the same problem from different directions. This is a fundamental observation common to most Markov decision process analysis. The following proof is due to [72].

Proof For any feasible vector J^l of (2.39) and any policy ν^l , we can combine the constraints of (2.39) (since $\nu^l(i, \cdot)$ is a probability distribution) to obtain:

$$J^l(i) \leq g^l(i, (\nu^l, \mu^{-l})) + \sum_{j \neq \tau} p(i, j; (\nu^l, \mu^{-l})) J^l(j), \quad (2.42)$$

That is, $J^l \leq \mathbf{g}^l(\nu^l, \mu^{-l}) + P(\nu^l, \mu^{-l})J^l$. From Theorem 2.2.1, it follows that the solution J^{*l} of (2.39) satisfies

$$J^{*l} \leq \mathbf{v}^l(\nu^l, \mu^{-l}) \text{ for all } \nu^l. \quad (2.43)$$

That the linear program (2.40) is dual to (2.39) implies the following *complementary slackness* condition on the solutions of (2.39) and (2.40) (see [72], Appendix G):

$$\pi^{*l}(i, u) > 0 \Rightarrow J^{*l}(i) = g^l(i, (u, \mu^{-l})) + \sum_{j \neq \tau} p(i, j; (u, \mu^{-l})) J^{*l}(j). \quad (2.44)$$

Since $\mu^{*l}(i, u) > 0$ if and only if $\pi^{*l}(i, u) > 0$, multiplying (2.44) by $\mu^{*l}(i, u)$ and summing over $u \in \mathbf{U}^l$ yields

$$J^{*l}(i) = g^l(i, (\mu^{*l}, \mu^{-l})) + \sum_{j \neq \tau} p(i, j; (\mu^{*l}, \mu^{-l})) J^{*l}(j). \quad (2.45)$$

That is, $J^{*l} = \mathbf{g}^l(\mu^{*l}, \mu^{-l}) + P(\mu^{*l}, \mu^{-l})J^{*l} = \mathbf{v}^l(\mu^{*l}, \mu^{-l})$. Therefore, policy μ^{*l} has cost J^{*l} , which is optimal by (2.43). A few details are omitted. ■

Proposition 2.4.1 presents the primal-dual linear programming method for solving a stochastic shortest path problem. The fundamental concept of duality arises often in optimization, and helps to provide much insight into the problem at hand. Moreover, knowing the solution to a linear programming problem automatically tells us much about the solution to the dual problem. For example, if (2.40) is solved, the solution to (2.39) is readily obtained through (2.45). Note that (2.45) is critical; it shows that μ^{*l} implements the optimal policy.

In Proposition 2.4.1, $J^{*l}(i)$ gives the total expected cost to player l starting from state i and following the optimal policy. $\pi^{*l}(i, u)$ gives the expected number of times the threat state is i and Player l uses control u . It follows that $\sum_u \pi^{*l}(i, u)$ represents the expected number of times state i is visited, so the policy μ^{*l} implements π^{*l} by randomizing amongst control choices in an appropriate way.

The objective of (2.40) has a natural interpretation in terms of minimizing the total ex-

pected cost. Its constraints ensure feasibility, by requiring that π is a stationary occupation measure for the induced Markov chain. Optimization problem (2.39) is less intuitive. The values $J^l(i)$ represent the total expected cost in state i , and the constraint reflects the fact that these costs must be minimized by the choice of control. The maximization objective ensures that the $J^l(i)$ do in fact represent the cost associated with some policy, ruling out trivial solutions.

If a single platform has control over all countermeasures, then the Proposition 2.4.1 yields an optimal centralized policy. This corresponds the case where all agents yield their autonomy to a central authority, which dictates actions in real time. As previously mentioned, such a solution is expensive to implement, and leaves the task group vulnerable in case of communication breakdown. Hence the need for decentralized, game-theoretic solutions.

2.4.2 Exploiting Transient Classes in the MDP

When the objective is to maximize the minimum distance, the state space augmentation of Section 2.3.4 introduces special properties into the Markov decision process, which we may take advantage of. In this section we assume, for simplicity, that each platform shares the common objective of maximizing the minimum distance to a target set G , and show how to exploit the existence of transient subsets of \mathbf{S} to increase computational efficiency. Transience may also arise in other ways (see Section 2.6.2), in which case a similar exploit may be used. The ideas of this section are based on the following lemma:

Lemma 2.4.1 *Suppose the state space \mathbf{S} of a transient Markov decision process can be subdivided into N transient classes, $\{\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_N\}$, and one absorbing state τ , such that*

$$p(i, j; u) = 0 \quad \forall u \in U(i) \text{ if } i \in \mathbf{S}_k, \quad j \in \mathbf{S}_l, \text{ and } k < l. \quad (2.46)$$

Define $\mathcal{S}_n = \{\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_n, \tau\}$. Then the optimal policy μ^ for the entire problem is equivalent to the optimal policy $\mu_{\mathcal{S}_n}^*$ obtained by considering the Markov decision process only on \mathcal{S}_n .*

Equation (2.46) denotes a requirement for a steady one-way dynamic in the Markov process; once the process reaches a set \mathbf{S}_k , it is impossible to go back to any state that belongs to a “higher” set. Thus, for planning purposes, a decision maker faced with a state in \mathbf{S}_k can ignore these higher states. For example, in the state space of Section 2.3.4, if a missile threat has reached a historic minimum distance of 1000 metres to the task group, it is impossible that in the future it will reach a historic minimum distance of 2000 metres, since

the historic minimum distance is nonincreasing.

Proof This is simply a consequence of Bellman's principle of optimality [64]. Consider the problem on \mathbf{S} . We have $\mathbf{v}^l(i, \mu^*) \leq \mathbf{v}^l(i, \mu_{\mathcal{S}_n}^*)$ for $i \in \mathbf{S}$ by optimality. Similarly, for the subproblem on \mathcal{S}_n , define the total expected cost $\mathbf{v}_{\mathcal{S}_n}^l(i, \mu)$ (with μ restricted to \mathcal{S}_n if appropriate). Again by optimality we have $\mathbf{v}_{\mathcal{S}_n}^l(i, \mu_{\mathcal{S}_n}^*) \leq \mathbf{v}_{\mathcal{S}_n}^l(i, \mu^*)$ for $i \in \mathcal{S}_n$. Now, by Theorem 2.2.1, we have $\mathbf{v}_{\mathcal{S}_n}^l(\mu^*) = \mathbf{g}_{\mathcal{S}_n}^l(\mu^*) + P_{\mathcal{S}_n}(\mu^*)\mathbf{v}_{\mathcal{S}_n}^l(\mu^*)$, where $\mathbf{g}_{\mathcal{S}_n}^l$ and $P_{\mathcal{S}_n}$ are just \mathbf{g}^l and P suitably restricted to \mathcal{S}_n . That is, for $i \in \mathcal{S}_n$,

$$\mathbf{v}_{\mathcal{S}_n}^l(i, \mu^*) = g^l(i, \mu^*) + \sum_{j \in \mathcal{S}_n} p(i, j, \mu^*) \mathbf{v}_{\mathcal{S}_n}^l(j, \mu^*) \quad (2.47)$$

$$= g^l(i, \mu^*) + \sum_{j \in \mathbf{S}} p(i, j, \mu^*) \mathbf{v}_{\mathcal{S}_n}^l(j, \mu^*), \quad (2.48)$$

where the last equality follows by hypothesis. By uniqueness of the total expected cost, it follows that $\mathbf{v}_{\mathcal{S}_n}^l(i, \mu^*) = \mathbf{v}^l(i, \mu^*)$ for all $i \in \mathcal{S}_n$. Repeating the argument for $\mathbf{v}_{\mathcal{S}_n}^l(i, \mu_{\mathcal{S}_n}^*)$, we have $\mathbf{v}^l(i, \mu_{\mathcal{S}_n}^*) \leq \mathbf{v}^l(i, \mu^*)$ for all $i \in \mathcal{S}_n$. Therefore $\mathbf{v}^l(i, \mu_{\mathcal{S}_n}^*) = \mathbf{v}^l(i, \mu^*)$ on \mathcal{S}_n . Since two policies are equivalent if they have the same cost, the proof is completed. ■

Since the previous minimum distance of the missile process of Section 2.3.4 is non-increasing, each set $\{s \in \mathbf{S} : d(s, G) = k\}$ constitutes a transient class for the process, and each set $\{s \in \mathbf{S} : d(s, G) < k\}$ likewise constitutes an absorbing class. Lemma 2.4.1 suggests a recursive method for computing an optimal policy when faced with these transient classes. First, we find an optimal policy over states such that $d(i, G) = 1$. Then we consider the problem over states such that $d(i, G) \leq 2$; since the optimal policy and costs is fixed for some states by Lemma 2.4.1, we may simply substitute these values into the new optimization problem. Continuing in this fashion we compute optimal policies for all states up to the maximum possible distance $d(i, G) = N$. This is summarized in the following corollary:

Corollary 2.4.1 *Suppose the Markov decision process for a single platform in the missile deflection problem can be divided into transient classes as in Lemma 2.4.1. Then the optimal policy can be computed via the following algorithmic sequence of linear programs:*

- Initialize $n = 1$.
- For all $i \in \mathbf{S}_n$, set

$$g^{*l}(i, u) = g^l(i, u) + \sum_{m < n} \sum_{j \in \mathbf{S}_m} p(i, j; (u, \mu^{-l})) \mathbf{v}^{*l}(j).$$

- Solve the linear program:

$$\begin{aligned}
 & \text{minimize} \quad \sum_{i \in \mathbf{S}_n, u \in U^l(i)} g^{*l}(i, (u, \mu^{-l})) \pi^l(i, u) \\
 & \text{subject to } (\forall j \in \mathbf{S}_n) \\
 & \quad \sum_{u \in U^l(j)} \pi^l(j, u) = \sum_{i \in \mathbf{S}_n, u \in U^l(j)} p(i, j; (u, \mu^{-l})) \pi^l(i, u) + \alpha(j), \\
 & \quad \pi^l(j, u) \geq 0 \quad \forall u \in U^l(j).
 \end{aligned} \tag{2.49}$$

- For $j \in \mathbf{S}_n$, compute $\mu^{*l}(j, u)$ as in (2.41), and set

$$\mathbf{v}^{*l}(j) = \mathbf{v}^l(\mu^{*l}, \mu^{-l})$$

as the solution to (2.45).

- Increment n and continue until $n = N + 1$.

Proof For $n = 1$, (2.49) directly gives π^{*l} on \mathbf{S}_1 by Lemma 2.4.1 and (2.40), since $\mathbf{S}_1 = \mathcal{S}_1$. Thus (2.41) and (2.45) also yield optimal $\mu^{*l}(j, u)$ and $\mathbf{v}^{*l}(j)$, respectively. For $n > 1$, assume that we have the optimal costs $\mathbf{v}^{*l}(j)$ for $j \in \mathcal{S}_{n-1}$. Consider (2.39) restricted to \mathbf{S}_n . Since Lemma 2.4.1 fixes $J^{*l}(j) = \mathbf{v}^{*l}(j)$ for $j \in \mathcal{S}_{n-1}$, we may rewrite this as:

$$\begin{aligned}
 & \text{maximize} \quad \sum_{i \in \mathbf{S}_n} \alpha(i) J^l(i) + \sum_{i \in \mathcal{S}_{n-1}} \alpha(i) \mathbf{v}^{*l}(i) \\
 & \text{subject to } (\forall u \in U^l(i)) \\
 & \quad J^l(i) \leq g^l(i, (u, \mu^{-l})) + \sum_{m < n} \sum_{j \in \mathbf{S}_m} p(i, j; (u, \mu^{-l})) \mathbf{v}^{*l}(j) + \sum_{j \in \mathbf{S}_n} p(i, j; (u, \mu^{-l})) J^l(j), \quad i \in \mathbf{S}_n \\
 & \quad \mathbf{v}^{*l}(i) \leq g^l(i, (u, \mu^{-l})) + \sum_{m < n} \sum_{j \in \mathbf{S}_m} p(i, j; (u, \mu^{-l})) \mathbf{v}^{*l}(j), \quad i \in \mathcal{S}_{n-1}, i \neq \tau
 \end{aligned} \tag{2.50}$$

Note that we have set the appropriate $p(i, j; u) = 0$ in the second constraint; it is then satisfied (with equality) by definition, so it may be dropped. In addition, the second sum in the objective is constant, so it may also be dropped. Substituting $\mathbf{g}^{*l}(\cdot)$ into the first constraint gives us a linear program exclusively on \mathbf{S}_n , the solution of which gives the optimal costs on \mathbf{S}_n . Now, it can be shown that (2.49) is dual to (2.50), once the above omissions and substitutions are made. We may now repeat the argument in Proposition

2.4.1: the complementary slackness condition can be used to show that μ^{*l} , obtained from (2.49), implements a policy with costs J^{*l} obtained from (2.50). Thus (2.49) computes the optimal policy on \mathbf{S}_n . The details are left for the reader. ■

Corollary 2.4.1 offers a significant reduction in computational complexity. First, note that obtaining $\mathbf{v}^{*l}(j)$ via (2.45) at each iteration may not be necessary, since many primal-dual methods for linear programming automatically produce the $\mathbf{v}^{*l}(j)$ as a by-product of calculation. Next, suppose each S_N is equal in size, then instead of solving one linear program with $N|S| \times U$ decision variables, we solve N linear programs with $|S| \times U$ decision variables. Under the best known linear programming algorithms, the number of calculations is thus reduced from order $(N|S| \times U)^{3.5}$ to order $N \times (|S| \times U)^{3.5}$, a computational savings of order $N^{2.5}$. This is important for the objective of Section 2.3.4, since we initially increased the size of the state space dramatically. The results here indicate that this increase isn't so bad; the complexity only increases linearly with the number of different distances considered.

2.4.3 Nash Equilibrium Solution for Multiple Ships

The computation above relies on the assumption that μ^{-l} is known and held fixed, but when all agents are simultaneously attempting to perform optimally, the situation is much more complex. In this case a reasonable option is to find and execute stationary policies which are in Nash equilibrium.

Recall the Nash equilibrium from Definition 2.2.3. Using Theorem 2.2.1, we rewrite this as follows:

Definition 2.4.1 *A Nash equilibrium in stationary policies is a joint policy μ such that for each player l and all $\nu \in \Delta(\mathbf{U}^l)$,*

$$(I - P(\mu))^{-1} \mathbf{g}^l(\mu) \leq (I - P((\nu, \mu^{-l})))^{-1} \mathbf{g}^l((\nu, \mu^{-l})). \quad (2.51)$$

That is, μ is a collection of individual policies $\{\mu^l, l \in L\}$ that are optimal responses to each other.

Such Nash equilibrium policies always exist (see [72]), and are locally optimal but generally inferior to centralized (Pareto optimal) policies because they suffer from coordination issues, essentially due to their restriction to the space of product distributions. However, this restriction is beneficial insofar as it requires no communication during the execution of the policy, and effectively distributes computation among all agents.

Nash policies are based on the notion that each platform's actions are a "best response" to the others' actions, so computation is naturally viewed as an iterative process, with candidate policies being improved upon by one or more players at each iteration until a fixed equilibrium point is reached. Indeed, iteration is easily seen to lead to equilibrium in the purely collaborative case, as the following proposition shows.

Proposition 2.4.2 *Iterated sequential best reply leads to Nash equilibria in collaborative games.*

Proof Since the expected costs coincide for each platform, and since each iteration necessarily reduces these costs (which are bounded below by zero), Monotone Convergence immediately yields the result. ■

Proposition 2.4.2 suggests the following algorithm for calculating a Nash equilibrium in a purely collaborative game.

Algorithm 2.4.1 *Agents adopt initial policies μ_0^l and i is initialized to zero. Repeat the following:*

1. *For $l = 1, \dots, |L|$ repeat:*
2. *Agent l computes a stationary policy μ_{i+1}^l via optimization problem (2.40), assuming the others use policies*

$$(\mu_{i+1}^1, \dots, \mu_{i+1}^{l-1}, \mu_i^{l+1}, \dots, \mu_i^{|L|}).$$
3. *If $l = |L|$, check whether the resultant joint policy is a Nash equilibrium. If it is not, reset l , increment i , and continue.*

Although a Nash equilibrium policy does not require communication during implementation, its computation does require that players communicate before an attack situation, as in Algorithm 2.4.1. This *offline* communication is unavoidable in this type of game, but since a task group is not so tightly bound by time and communication constraints before a missile attack, such communication is feasible. A new Nash equilibrium needs to be negotiated only when the task group configuration, threat type, or defensive capabilities change.

That said, we observe that the efficiency of Algorithm 2.4.1 is rather poor, since players must continuously communicate their latest policies, and each player spends most of its time waiting for the others to finish their computations. To increase efficiency, players can

initially compute best responses in parallel until a suitably good policy is found, so that Algorithm 2.4.1 may be initialized close to equilibrium, so that fewer serial iterations are required.

In the non-collaborative case, the best reply dynamic is not guaranteed to converge to a Nash equilibrium, but may still yield acceptable results. Guaranteed methods for computing Nash equilibria in general sum stochastic games are still being developed. For example, stochastic tracing procedure using Homotopy theory has recently been developed in [74], while new nonlinear programming methods have been put forward in [83]. More established nonlinear programming methods, such as those in [75] and [67] are also useful, although not guaranteed to converge.

2.4.4 Pareto Optimality and Nash Equilibrium

The Nash equilibrium policy is optimal from each individual perspective, but what can be said about optimality from the collective point of view? The accepted measure of collective performance is Pareto optimality, which requires us to define *joint policies* as follows:

Definition 2.4.2 Let $\mathbf{U}(i) = \mathbf{U}^1(i) \times \mathbf{U}^2(i) \times \dots \times \mathbf{U}^L(i)$ be the vector space of possible joint actions given threat state i , and let $\Delta\mathbf{U}(i)$ represent the set of probability distributions over $\mathbf{U}(i)$. A joint policy μ is a function $\mu(i) : i \in \mathbf{S} \rightarrow \Delta(\mathbf{U}(i))$; for each $i \in \mathbf{S}$, the probability of deploying joint CM vector u is $\mu(i, u)$.

Definitions 2.4.2 and 2.2.1 are very similar. Moreover, policy profiles, as in Definition 2.2.2, are a subset of joint policies; they are simply joint policies restricted to a product space.

The concept of Pareto optimality, in general, can be stated as follows:

Definition 2.4.3 A joint policy $\mu \in \Delta(\mathbf{U})$ is Pareto optimal if there is no alternative joint policy ν that improves the utility of one individual without decreasing the utility of another. That is, if:

$$\mathbf{v}^l(i, \mu) \leq \mathbf{v}^l(i, \nu) \quad (2.52)$$

for all $i \in \mathbf{S}$, $l \in L$ and all $\nu \in \Delta(\mathbf{U})$.

A Nash equilibrium need not be Pareto optimal. In fact, in the presence of constraints, the Pareto optimal policy may not even belong to the set of policy profiles, so that no independently executable policies can implement it, simply for coordination reasons. (See the example in Section 2.6.1.) Thus, Pareto optimality may be out of reach of the class of independently executable policies.

We are thus led to consider the weaker property of Pareto optimality restricted to independently executable policies, as follows:

Definition 2.4.4 *A joint policy $\mu \in \Delta(\mathbf{U}^1) \times \Delta(\mathbf{U}^2) \times \dots \times \Delta(\mathbf{U}^l)$ is Pareto optimal among independently executable policies if there is no alternative independently executable policy ν that improves the utility of one individual without decreasing the utility of another. That is, if:*

$$\mathbf{v}^l(i, \mu) \leq \mathbf{v}^l(i, \nu) \quad (2.53)$$

for all $i \in \mathbf{S}$, $l \in L$ and all $\nu \in \Delta(\mathbf{U}^1) \times \Delta(\mathbf{U}^2) \times \dots \times \Delta(\mathbf{U}^l)$.

This definition leads us to the following proposition:

Proposition 2.4.3 *In the collaborative game case, at least one Nash equilibrium is Pareto optimal among independently executable policies.*

Proof Since all player objectives are the same, Pareto optimality is equivalent to the condition that no policy exists that improves the common utility. Consider the independently executable policy with the highest utility (such a policy exists since the utility is bounded). This policy is Pareto optimal among independently executable policies by definition. Also, since the utility is maximal, no unilateral deviation can improve any player's utility, hence the policy is also a Nash equilibrium. ■

When the game is not collaborative, a pricing scheme may be useful to promote Pareto optimality. Essentially, another phase would be added to the problem in which agents would choose which targets in G to include in their target sets $G(l)$. The targets would represent assets which would be traded on an open market by the agents. Once the targets were set, a Nash equilibrium would be computed. This process is continued with further rounds of bidding and policy computation. It is well-known that such free market schemes lead to Pareto optimality. However, this is beyond the scope of the current chapter, and is relegated to future work.

2.5 Introducing CM Limitations via Constraints

So far, we have formulated the missile deflection problem under just one simple constraint type; i.e. that l 's CM choices are restricted by the threat state to $U^l(i)$, which may be a subset of all available CM choices. This may be used to keep ships from interfering with each other by violating doctrinal rules (e.g. deploying CMs in undesired directions), but cannot

handle dynamic constraints. For example, we may wish to model limitations on the number of countermeasures available, or limitations on allowable sequences of countermeasures. The purpose of this section is to outline methods for adding such constraints.

We outline three such approaches. The first two approaches yield an optimal policy when a limit is placed on the number of times each type of CM may be used, while the third approach yields an optimal policy when a limit is placed on how CM choices may be used in sequence. Among these, the first approach yields a policy in which the actual number of uses of each CM type is bounded, while the second approach yields a policy in which only the average number of uses is bounded. Despite the weaker nature of the second approach, it is still preferred since the state space is more manageable. All three approaches can be combined as desired for use in the game-theoretic algorithms.

2.5.1 Limiting CM availability via State Space Augmentation

For each platform l , partition $\cup_{i \in \mathbf{S}} U^l(i)$ into $\{\mathbf{u}_k^l : k = 1, 2, \dots, I(l)\}$. Each \mathbf{u}_k^l corresponds to a certain type of CM available to platform l , with set limit $\eta(\mathbf{u}_k^l)$ on the number of times it may be deployed. To reflect the depletion of countermeasures, we perform a state space augmentation similar to (2.34) in Section 2.3.4.

As in that section, define a new stochastic process on the state space

$$\mathbf{S} := \mathbf{S} \times \mathcal{D}^1 \times \dots \times \mathcal{D}^L. \quad (2.54)$$

where each \mathcal{D}^l is a vector-valued set

$$\mathcal{D}^l = \{d \in \mathbb{Z}^{I(l)} : 0 \leq d \leq (\eta(\mathbf{u}_1^l), \dots, \eta(\mathbf{u}_{I(l)}^l))\} \quad (2.55)$$

The k^{th} component of \mathcal{D}^l represents the number of CMs of type k left to platform l .

Denote the original state space by $\bar{\mathbf{S}}$, the corresponding states by \bar{i} , and the original transition probabilities in (2.27) by $\bar{p}_{\bar{i}\bar{j}}(u)$. The transition probabilities on the new state space \mathbf{S} are:

$$p(i, j; u) = \bar{p}(\bar{i}, \bar{j}, u) \prod_{l \in L} I\left(u^l \in \mathbf{u}_k^l \text{ and } j_{\mathcal{D}^l} = i_{\mathcal{D}^l} - e_k\right), \quad (2.56)$$

where $i_{\mathcal{D}^l}$ represents the portion of i restricted to \mathcal{D}^l , and e_k is an appropriate unit vector. The quantity $i_{\mathcal{D}^l}$ gives the number of countermeasures of each type left to player l in state i , so the indicators in (2.56) pick out the state j that represents the fact that the appropriate CM reserve of Player l is decreased by one. The stochastic process so defined

is non-increasing in each new dimension.

We also redefine the allowable CM choices by setting

$$U^l(i) := U^l(\bar{i}) \cap \bigcup_{k \in K(i)} \mathbf{u}_k^l, \quad (2.57)$$

where

$$K(i) = \{k \in \mathbb{Z} : k^{th} \text{ component of } i_{\mathcal{D}^l} > 0\}. \quad (2.58)$$

That is, CM choices are restricted to those that have not yet run out. Since the optimal policy follows the principles of dynamic programming, Optimization problem (2.40) will automatically account for the limited supply of countermeasures when applied to the augmented problem. That is, μ^l will yield an optimal joint or best response policy that minimizes the cost while respecting the CM limitations. Furthermore, the newly defined state space can clearly be subdivided into transient classes, in the same manner as the augmented process incorporating the previous minimum distance was in Section 2.4.2. Hence there exists a sequence of linear programs that yields the optimal solution to this problem.

Remark As the $\eta(\mathbf{u}_i^l)$ increases, we expect the constrained optimal policy to approach the unconstrained policy in a smooth manner. The size of the augmented state space also increases, however, so that the problem becomes more complex. If the optimal policy does indeed change smoothly, we may approximate the optimal policies for various values of the $\eta(\mathbf{u}_i^l)$ by first computing both the unconstrained optimal policy and the constrained optimal policy for $\eta(\mathbf{u}_i^l)$ all small (1 or 2 perhaps), and then using some weighted mixture of these two optimal policies for all other values of the $\eta(\mathbf{u}_i^l)$. A complete investigation of this method is beyond the scope of this chapter.

2.5.2 Limiting CM Directly in the Linear Program

In this section, we place an upper bound on the *expected* number of uses of different CM types, instead of the actual number of uses. Define \mathbf{u}_i^l as in Section 2.5.1, and place a limit $\eta(\mathbf{u}_i^l)$ on the *expected* number of times each CM type is used. We begin with the following lemma.

Lemma 2.5.1 *In the context of Optimization Problem (2.40),*

$$\sum_{u^l \in \mathbf{u}_i^l} \sum_{j \in \mathbf{S}} \pi(j, u^l)$$

is equal to the expected number of times platform l will use CM type i .

This follows given the interpretation of $\pi(i, u)$ as the expected number of times CM u will be used in state i . However, we give a formal proof of this assertion.

Proof We drop the superscripts separating platforms for brevity. For a fixed CM u , define row vectors α , π_u , and μ_u over \mathbf{S} with i^{th} entry $\alpha(i)$, $\pi(i, u)$, and $\mu(i, u)$, respectively. Let x^T represent the transpose of x . Let $A_{\mu_u} = \text{diag}(\mu_u)$ be a diagonal matrix with entries $A_{\mu_u}(i, i) = \mu(i, u)$; the inverse has entries $A_{\mu_u}^{-1}(i, i) = 1/\mu(i, u)$. (Actually this is a formalization, since $\mu(i, u)$ may zero, but the argument holds if we set $0/0 = 0$.) From (2.41), the definition of μ , we have that $\pi(i, u)/\mu(i, u) = \sum_z \pi(i, z)$ for all i, u , or,

$$\pi_u A_{\mu_u}^{-1} = \sum_z \pi_z, \text{ for all } u. \quad (2.59)$$

Letting P_u represent the transition matrix under control u , we may write the constraint of (2.40) as

$$\begin{aligned} \alpha &= \sum_u \pi_u (I - P_u) \\ &= \sum_u \pi_u A_{\mu_u}^{-1} A_{\mu_u} (I - P_u) \\ &= \left(\sum_z \pi_z \right) \sum_u (A_{\mu_u} - A_{\mu_u} P_u) \\ &= \left(\sum_z \pi_z \right) (I - P(\mu)), \end{aligned} \quad (2.60)$$

where the last equality follows since $\sum_u A_{\mu_u}(i, i) = \sum_u \mu(i, u) = 1$, and $\sum_u A_{\mu_u}(i, :) P_u(:, j) = \sum_u \mu(i, u) p(i, j; u) = p(i, j; \mu)$. Now, we may find the desired expectation,

$$\begin{aligned} E[\text{times } u_r \text{ is used}] &= \sum_j f_{tc}(\alpha, \mu; j) \mu(j, u_r) \\ &= \alpha (I - P(\mu))^{-1} \mu^T(u_r) \\ &= \left(\sum_z \pi(z) \right) (I - P(\mu)) (I - P(\mu))^{-1} \mu^T(u_r) \\ &= \sum_z \sum_i \pi(i, z) \mu(i, u_r) \end{aligned}$$

$$\begin{aligned}
 &= \sum_z \sum_i \pi(i, z) \frac{\pi(i, u_r)}{\sum_z \pi(i, z)} \\
 &= \sum_i \pi(i, u_r).
 \end{aligned} \tag{2.61}$$

Summing over all controls u_r in a given class \mathbf{u}_i^l gives the result. ■

In light of Lemma 2.5.1, adding the constraints

$$\sum_{u^l \in \mathbf{u}_i^l, j \in \mathbf{S}} \pi(j, u^l) < \eta(\mathbf{u}_i^l), \quad \forall l \in L, i = 1, \dots, I^l \tag{2.62}$$

to Optimization Problem (2.40) should yield an optimal policy that restricts the average use of each CM type appropriately.

2.5.3 Constraining CM Sequences

The final constraint addresses the possibility that the allowable CM choices at time $k + 1$ may be limited by the choice at time k . Such a constraint may arise due to mechanical reasons, for example if the countermeasures are decoys which can only move at a limited rate, or if the countermeasure consists of a one-time “launch” command. The constraint may also be used to limit the rate of change in false target positions in light of the fact that a missile would most likely recognize any erratically moving target as false, thus negating its effectiveness.

We include this constraint type by incorporating the current CM action as part of the state space. That is, redefine $\mathbf{S} := \mathbf{S} \times \mathbf{U}$, so that state $i \in \mathbf{S}$ reflects not only the missile state, but also the current joint CM action in $U(i)$. We then redefine each set $U^l(i)$ to be the new, restricted set of actions available to platform l given that the current action is incorporated into state i . Once these spaces are defined, the optimal solution is computed exactly as in the unconstrained case.

Example (Chaff.) For each possible chaff launch, the control choices can be thought of as belonging to the set

$$\{\text{Hold, Launch, Null}\}. \tag{2.63}$$

For any state $i = (i_x, i_z, i_u)$, where i_u is the control choice in state i , simply set

$$U^l(i) = \begin{cases} \{\text{Hold, Launch}\}, & i_u = \text{Hold}, \\ \{\text{Null}\}, & i_u \in \{\text{Launch, Null}\}. \end{cases} \tag{2.64}$$

Running the optimization algorithm over the new state space $\mathbf{S} := \mathbf{S} \times \mathbf{U}$ will yield a joint policy in which agents are aware of and restricted to one-time “launch” controls for chaff.

Example (Decoy velocity constraint.) For any state $i = (i_s, i_u)$, where i_u is the control in state i specifying the physical position of a decoy, define $d(a, b)$ to be the distance between two decoy positions and set

$$U^l(i) = \{j_u : d(i_u, j_u) \leq d_{\max}\}. \quad (2.65)$$

That is, future decoy positions are restricted to be sufficiently close to the present decoy position. Running the optimization algorithm over the new state space $\mathbf{S} := \mathbf{S} \times \mathbf{U}$ will again yield a joint policy in which agents are aware of and restricted to rate-limited changes in decoy positions.

The cost of such a formulation is an exponential increase in the size of the state space. However, this is at least partially offset by an increased sparsity in the transition matrices, since many state-control combinations will never occur. In addition, the number of possible controls in each state is reduced substantially in the new formulation, reducing the computational complexity involved in calculating an optimal policy significantly.

2.6 Numerical Results

In this section we investigate two simple but illustrative missile deflection examples. In the first example, two ships jointly influence the motion of a randomly moving missile by presenting a single false target. In the second, three ships jointly influence the targeting system of a missile by slowly introducing angle errors through electromagnetic jamming.

2.6.1 Maximizing Minimum Approach of a Missile with Simple Targeting

In the first example, two stationary ships are threatened by a single missile moving on an 11×11 battlespace $B = \{(x, y) \in \mathbb{Z}^2 : -5 \leq x, y \leq 5\}$. The two ships are located at $(0, 0)$ and $(0, -2)$, with $G = G(1) = G(2) = \{(0, 0), (0, -2)\}$. We seek a Nash equilibrium policy with the objective of Section 2.3.4; maximizing the minimum distance from the missile to the pair of ships.

The CM capabilities involve the joint control of a single decoy. We set $U^1 = U^2 = \{(1, 0), (-1, 0), (0, 1), (0, -1)\}$, indicating the x - y displacement of the decoy from the Origin

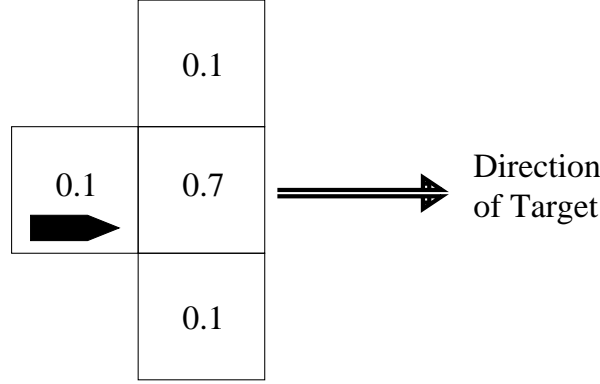


Figure 2.2: Missile transition probabilities $p_y^m(i, j; z, u)$ relative to target position z .

by each ship. The joint control is a superposition; $U_n = U_n^1 + U_n^2$. The missile may target either the decoy or Ship One, hence $T = \{(x, y) \in \mathcal{Z}^2 : |x| + |y| \leq 2\}$.

Define $f(i, z)$ to be the next grid point along the shortest path between i and z . Define $g(i, z) = (0, 1)$ or $(1, 0)$, depending on whether $|i_x - z_x| > |i_y - z_y|$ or vice-versa. Then the missile dynamics are given by:

$$p_z^m(i, j; y, u) = \begin{cases} 0.2, & j = (0, 0), \\ 0.8, & j = u. \end{cases} \quad (2.66)$$

$$p_y^m(i, \tau; z, u) = \begin{cases} 0.5, & i = z, \\ 1, & i = (0, 0) \text{ or } (0, -2), \\ 0.02, & \text{otherwise} . \end{cases} \quad (2.67)$$

$$p_y^m(i, j; z, u) = (1 - p_y^m(i, \tau; z, u)) \cdot \begin{cases} 0.7, & j = f(i, z), \\ 0.1, & j = f(i, z) \pm g(i, z), \\ 0.1, & j = i. \end{cases} \quad (2.68)$$

See Figure 2.2 for an illustration of the last equation. The missile targets either Ship One or the decoy; Ship Two is not targeted, but may still be hit if reached. At time zero, the missile appears randomly on the edge of the grid and subsequently moves according to the above dynamics, as influenced by the chosen CM policy.

Based on the transition probabilities derived from the above dynamics, Algorithm 2.4.1 was used to compute a stationary Nash equilibrium policy. This policy is characterized as follows. For states $(x, y \in G)$, (the shaded “area of active control” in the figures), the agents attempt to lure the missile as far away from Ship One as possible by placing the perceived

decoy at the nearest state such that $|x| = 2$ or $|y| = 2$. This maximizes the probability that the missile will hit the decoy, and minimizes the probability that it will hit a ship.

Since there is no coordination, this is not always possible. For example, when the missile is at $(1, 1)$, Ship One may decide to move the decoy east while Ship Two may move the perceived position north. Each has attempted to lure the missiles to $(2, 0)$ and $(0, 2)$, respectively, but the outcome is a decoy position of $(1, 1)$. This coordination problem cannot be resolved without communication or convention.

For states outside the active control area, the agents randomly choose CMs that cause the missile to move toward the nearest state such that $|x| = 2$ or $|y| = 2$. For example, if the missile is at $(5, 0)$, then the goal is to guide the missile to $(2, 0)$, which is accomplished by placing the decoy at $(x, 0)$, $|x| \leq 2$.

This policy is quite reasonable. As a check, an optimal policy was also computed by assuming a single ship had control over both countermeasures. The Pareto optimal costs are shown in Figure 2.4. Except for the coordination problems associated with independently randomizing decisions, the Nash equilibrium policy was almost identical to the optimal one. Figure 2.3 shows the costs associated with the equilibrium policy, that is the expected difference between the initial and minimum distances achieved over the lifetime of the missile process. Figure 2.5 shows two sample missile paths from two separate simulations, along with the false target positions used at different times during the attack.

2.6.2 Minimizing Probability of Hit for a Missile with Forward Momentum

In the second example, three stationary ships equipped with CM lie at the edges of a task group. A single missile threatens the task group, and the objective of the three ships is to minimize the probability that the missile reaches any point in the interior of the triangle defined by their positions. This example has two simplifying features. First, the missile moves one unit from left to right at each step, so the threat terminates by a definite time, and only has one pass at the targets. Second, CM actions are independent in the sense that the jamming success probabilities of each ship are multiplied to determine the overall CM effects.

Define the battlespace $B = \{(x, y) \in \mathcal{Z}^2 : 1 \leq x, y \leq 11\}$ (see Figure 2.7). The true target set G is the shaded area in that figure. We deviate from our setup slightly and define the missile target set $T = \{(x, y) \in \mathcal{Z}^2 : x = 11, 1 \leq y \leq 11\}$. That is, we assume the missile target is along the right-hand edge of the state space, instead of in the task group. Define

3.50	3.51	3.46	3.38	3.33	3.32	3.33	3.38	3.46	3.51	3.50
3.53	2.62	2.61	2.52	2.43	2.41	2.43	2.52	2.61	2.62	3.53
3.53	2.65	1.71	1.66	1.52	1.47	1.52	1.66	1.71	2.65	3.53
3.48	2.63	1.75	0.78	0.66	0.46	0.66	0.78	1.75	2.63	3.48
3.40	2.53	1.67	0.97	0.24	0.24	0.24	0.97	1.67	2.53	3.40
3.37	2.47	1.51	0.48	0.25	0	0.25	0.48	1.51	2.47	3.37
3.37	2.47	1.54	0.68	0.25	0.28	0.25	0.68	1.54	2.47	3.37
3.41	2.54	1.69	0.80	0.21	0	0.21	0.80	1.69	2.54	3.41
3.49	2.63	1.74	0.84	0.29	0.60	0.29	0.84	1.74	2.63	3.49
3.54	2.66	1.74	0.89	1.12	1.40	1.12	0.89	1.74	2.66	3.54
3.54	2.63	1.77	1.87	2.01	2.20	2.01	1.87	1.77	2.63	3.54

Figure 2.3: Expected costs \mathbf{v}^l for the Nash equilibrium policy computed by Algorithm 2.4.1, arranged over a spatial grid of possible missile positions. The shaded area indicates the possible false target positions and the marked squares indicate ship positions $S1$ and $S2$. The values are the expected cost associated with the process when the missile is in each position, i.e. the difference between the initial and the expected minimum future distance.

3.48	3.49	3.45	3.38	3.33	3.32	3.33	3.38	3.45	3.49	3.48
3.49	2.60	2.59	2.52	2.43	2.41	2.43	2.52	2.59	2.60	3.49
3.45	2.59	1.69	1.65	1.52	1.46	1.52	1.65	1.69	2.59	3.45
3.38	2.52	1.65	0.77	0.65	0.46	0.65	0.77	1.65	2.52	3.38
3.33	2.43	1.52	0.65	0.24	0.23	0.24	0.65	1.52	2.43	3.33
3.33	2.41	1.47	0.46	0.24	0	0.24	0.46	1.47	2.41	3.33
3.33	2.43	1.53	0.66	0.25	0.28	0.25	0.66	1.53	2.43	3.33
3.39	2.53	1.67	0.78	0.21	0	0.21	0.78	1.67	2.53	3.39
3.47	2.62	1.73	0.83	0.28	0.60	0.28	0.83	1.73	2.62	3.47
3.53	2.65	1.73	0.88	1.11	1.40	1.11	0.88	1.73	2.65	3.53
3.53	2.61	1.76	1.86	2.00	2.19	2.00	1.86	1.76	2.61	3.53

Figure 2.4: Expected costs \mathbf{v}^l for the Pareto optimal policy for the first example, arranged over a spatial grid of possible missile positions. The shaded area indicates the possible false target positions and the marked squares indicate ship positions $S1$ and $S2$. The values are the expected cost associated with the process when the missile is in each position, i.e. the difference between the initial and the expected minimum future distance.

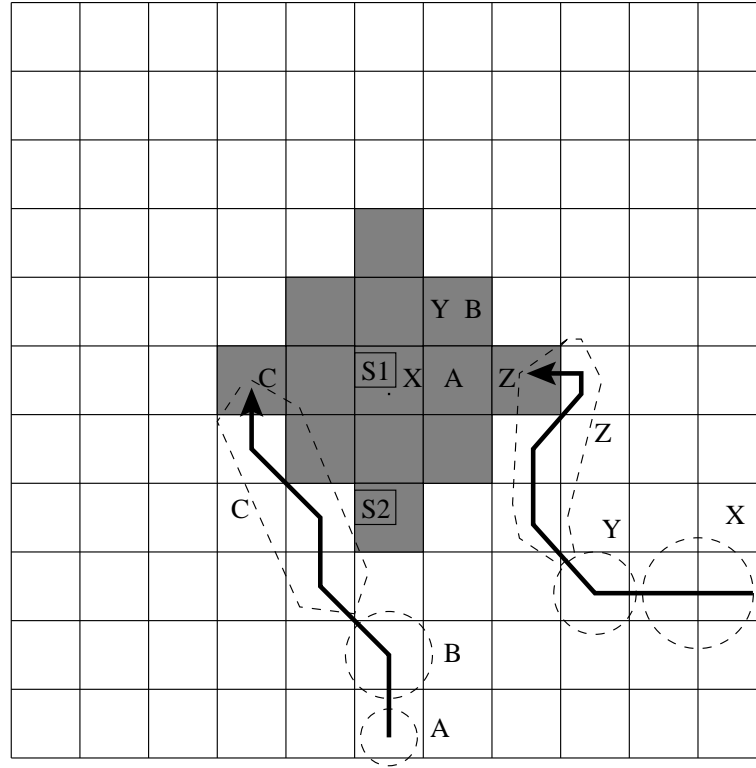


Figure 2.5: Sample missile and CM paths under the Nash policy from two initial points. The grid represents the space of possible missile positions. The shaded area indicates the possible false target positions and the marked squares indicate ship positions. The false target positions applied for each portion of the sample missile trajectories (the bold arrows) are denoted by A,B,C for the first simulated trajectory, and by X,Y,Z for the second simulated trajectory. These letters denote both the false target positions and the portions of the missile trajectories for which they were employed.

the target angle θ as in Figure 2.6. That is, for missile state $(Y_n, Z_n) = ((a_x, a_y), (11, b_y))$,

$$\cos(\theta(Y_n, Z_n)) = \sqrt{(11 - a_x)^2} / \sqrt{(11 - a_x)^2 + (b_y - a_y)^2}.$$

The physical transition probabilities are given by:

$$p_y^m(i, j; z, u) = \begin{cases} \cos(2\theta(i, z)), & j = (i_x + 1, i_y), \\ 1 - \cos(2\theta(i, z)), & j = (i_x + 1, i_y + \text{sign}(\theta(i, z))) \\ 1, & j = \tau, i_x = 11 \text{ or } i \in G. \end{cases} \quad (2.69)$$

The target angle update is the point at which the missile is vulnerable to jamming. At each time index, the agents are able to independently attempt to shift the equivalent target position up or down by one unit. Define the probability of success of each CM attempt to be p_s ;

$$p_s = \text{Pr}(\text{CM action of ship } l \text{ is effective at time index } k). \quad (2.70)$$

p_s is an empirical value based on factors such as missile type, environment, etc. The CM control choices for each ship are $\mathbf{U}^l = \{-1, 0, 1\}$, in keeping with the discussion on constrained CM sequences in Section 2.5.3. We may quantify this as follows. Define e_i to be the i^{th} basis vector in eleven dimensions (since there are eleven target positions). Then we may define a probability distribution:

$$F(i, u) = \sum_{A \subset \{1, 2, 3\}} p_s^{|A|} (1 - p_s)^{3 - |A|} e_{(i + \sum_{l \in A} u^l)}, \quad (2.71)$$

where $|A|$ means the number of components of A , and the sum is over all possible subsets A of successful players. This gives the transition probabilities; $p_z^m((11, i_y), (11, j_y); y, u)$ is the j_y^{th} component of $F(i_y, u)$.

As before, a Nash equilibrium solution was obtained, using the minimal probability of hit objective of Section 2.3.4. We present a representative trajectory in Figure 2.7 for CM success probability $p_s = 0.6$. The probability of hit for this case, when the missile is initially distributed uniformly along the left edge of the state space $Z_1 = (11, 6)$, is 0.13, whereas the probability of hit is 0.91 if no countermeasures are used.

The joint Nash equilibrium for the second example also appears reasonable. Roughly speaking, the agents coordinate their efforts to drive the target to the side during the initial stages of the attack, and then as the danger level decreases, they become more indifferent as to their CM choices. The iterated best response algorithm leads to an independent

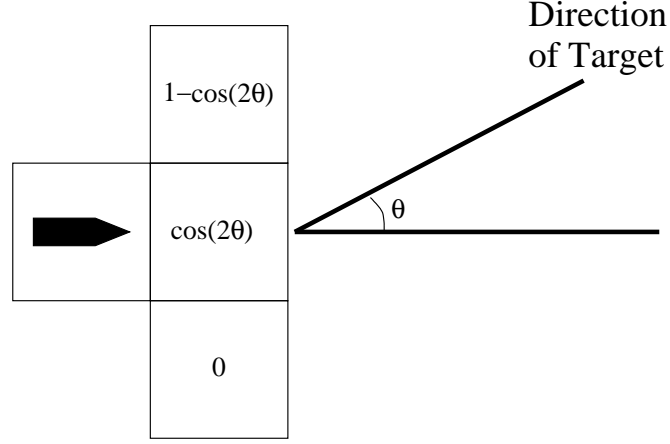


Figure 2.6: Missile transition probabilities $p_y^m(i, j, z, u)$ relative to target position z for second example. The probabilities are shown for a positive angle θ .

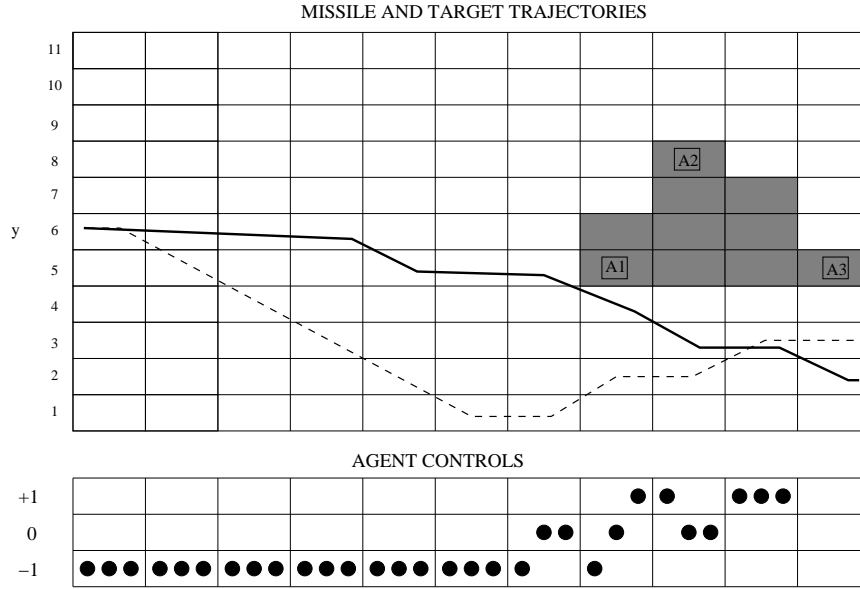


Figure 2.7: Sample trajectory of missile, target location, and CM choices under Nash policy for $p_s = 0.6$. The solid line indicates the missile flight path Y_n , while the broken line indicates the y-axis position of the missile's current target Z_n . The control choices of all three agents are depicted below the graph for each missile position. The shaded area represents the task group, to be defended, with the three agents A1, A2, A3 at its vertices.

synchronization of actions, for example all agents will attempt to move the target down at the beginning of the attack, as opposed to one ship working to cancel the efforts of another.

Sensitivity Analysis of Nash Equilibrium to Missile Dynamics The missile characteristics may not be perfectly known by the controller, in which case one may wish to measure how the performance of a joint policy varies with modeling errors. In this section, we introduce an error between the true CM success probability p_s , defined in (2.70) as the probability of successfully influencing the missile target, and the estimate of p_s used by agents when computing their optimal CM policies, \hat{p}_s . We then analyze the performance of Nash equilibrium policies under various values of p_s and \hat{p}_s .

In particular, Nash equilibrium policies were computed for values of $\hat{p}_s = 0.05, 0.1, \dots, 1.0$. These policies were then evaluated under models with true parameters $p_s = 0.05, 0.1, \dots, 1.0$. The resulting probability of hit, for an initial distribution as before (missile uniformly distributed along the left edge of the state space, with initial target $y = 6$), are presented in Figure 2.8.

The first characteristic to note is that the costs associated with the policies changes very little as the model parameter \hat{p}_s is varied slightly from the true parameter p_s . That is, the system is not particularly sensitive to small modeling errors. The exception to this is that the system appears to be relatively more sensitive to differences in \hat{p}_s between 0.15 and 0.25, where a small modeling error such as $\hat{p}_s = 0.25$ while $p_s = 0.20$ can result in a 10% difference between predicted and actual costs. In general though, small modeling errors result in negligible performance differences (i.e. the graph is flat in the direction of changing \hat{p}_s).

More interesting is the fact that the costs decrease as both p_s and \hat{p}_s increase, with the exception of the region where $\hat{p}_s > 0.95$. The reason for the decrease along the direction of p_s is clear: as the agents are better able to influence the missile's targeting system, the performance of the CM policies improve. The fact that costs decrease along the direction of \hat{p}_s as well is more surprising, indicating that the agents perform better when they are more confident about their actions. That is, if a ship computes a policy based on the assumption that it and its fellows have strongly effective CM, it will do better than if it computed its policy based on the true, weaker effectiveness. It is doubtful that this is a general phenomenon, and the appearance of it here is most likely due to the underlying independence of CM effects.

Nevertheless, the monotonicity suggests a method for computing improved Nash equilibrium policies in this example. Namely, the agents compute a Nash equilibrium policy

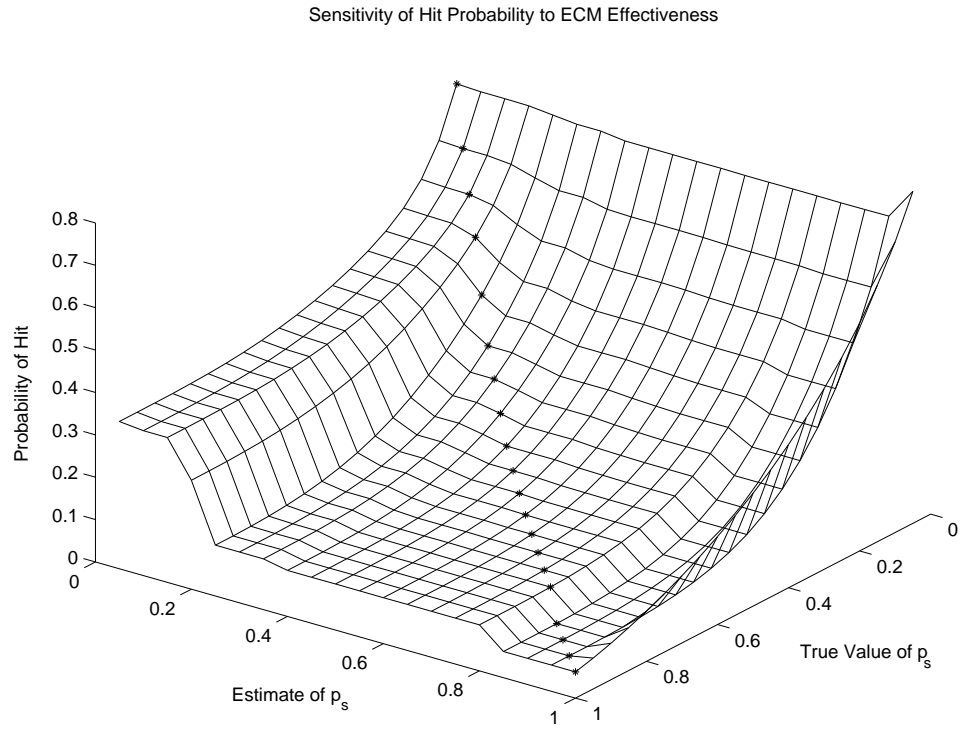


Figure 2.8: Performance of Nash equilibrium policies when p_s , the CM success probability as defined in (2.70), and its estimate, \hat{p}_s , vary. As p_s increases, CM becomes more effective and hence the probability of hit is reduced. As \hat{p}_s deviates from p_s , the CM policy and hence the probability of hit varies, although the true missile dynamics remain unchanged.

assuming an inflated value of \hat{p}_s , then use this as a starting point for the computation of a Nash equilibrium based on the true value $\hat{p}_s = p_s$. Since the iterated best response technique is monotonic, the resultant equilibrium policy will have a better performance than the original. Interestingly, when this procedure was carried out, it was found that policies based on inflated values of \hat{p}_s were already equilibrium policies for lesser values. It is suspected that this surprising fact is again due to the underlying independence of CM effects.

2.7 Conclusion

Decentralized missile deflection policies, computed offline, give task groups the ability to effectively defend against attacks even without real-time communication. This ability allows a task group to capitalize on the benefits of NCW while minimizing its vulnerability to attack on the infostructure that lies at its heart.

In this chapter, we formulated a decentralized missile deflection problem, including a discussion of missile and countermeasure dynamics and the development of appropriate defensive objectives. This formulation leads to an interpretation of the problem as a stochastic shortest path game, for which Nash equilibrium solutions are known to exist. Algorithms to compute a Nash equilibrium solution via linear programming and iterated best response were proposed and shown to be effective when player objectives coincide. Finally, numerical studies and sensitivity analysis were presented to show how these results can be applied.

Nash equilibrium solutions will in general be less effective than Pareto optimal solutions. However, in our simulations the degradation is minimal. In particular, when player objectives coincide and there are no extra constraints, there exists a pure optimal policy [81]. It follows that this optimal policy is also a pure Nash equilibrium. Therefore, it is no surprise that Nash policies perform quite well in the team games considered here.

Game theory presents a promising avenue of approach for solving complex decision problems in a decentralized fashion. By distributing the problem among several decision makers, the decision space is significantly reduced, so that near-optimal solutions can be computed with far less complexity than that required for an optimal centralized solution. The framework presented here applies in a very general setting. Provided that the missile dynamics are known and agreed upon, that each ship is known to have perfect observations of missile threats, and that ship capabilities and objectives are common knowledge, both globally optimal and Nash equilibrium solutions can in theory be computed for any threat/defense combination, for any set of player objectives.

Proof of Theorem 2.2.1

Proof As in [82], define a matrix C with entries $c_{ij} = f_{tc}(\delta_i \cdot \mu; j)$; the expected number of time periods that the Markov chain is in j given that it starts in i . Conditioning on the first transition out of i , we obtain

$$c_{ij} = I\{i = j\} + \sum_{k \neq \tau} p(i, k; \mu) c_{kj}, \quad (2.72)$$

where $I\{\cdot\}$ is the indicator function. Since the Markov chain induced by $P(\mu)$ is transient, this sum is finite. In matrix form we have

$$C = P(\mu)C + I, \quad (2.73)$$

$$C = (I - P(\mu))^{-1}. \quad (2.74)$$

Moreover, since

$$f_{tc}(\alpha \cdot \mu; j) = \sum_{i \in \mathbf{S}} f_{tc}(\delta_i \cdot \mu; j) \alpha(i) \quad (2.75)$$

by linearity, we have, for α a row vector on \mathbf{S} ,

$$f_{tc}(\alpha \cdot \mu; j) = \alpha C = \alpha (I - P(\mu))^{-1}. \quad (2.76)$$

Equation (2.4) and the definition of $\mathbf{g}^l(\mu)$ directly yield the result for $\mathbf{v}_\alpha^l(\mu)$. The result for $\mathbf{v}^l(\mu)$ follows by setting $\alpha = \delta_i$ for each $i \in \mathbf{S}$. Equation (2.9) follows directly by substituting the Taylor series expansion $(I - P(\mu))^{-1} = I + P(\mu) + (P(\mu))^2 + \dots$ into (2.7). ■

Glossary

Variable	Interpretation
M	Set of incoming missile threats.
L	Set of countermeasure (CM)-equipped ships (platforms).
$\mathbf{S}^m = \{B \times T, r, \tau\}$	State space of missile m ; {position \times target, initial, terminal}.
$(Y_n^m, Z_n^m) \in \mathbf{S}^m$	Missile m threat state; (position, target).
$G \subset T$; $G(l) \subset G$	Set of true targets; Set of true targets defended by platform l .
$U_n^l \in \mathbf{U}^l$; $U_n \in \mathbf{U}$	CM action of Platform l at time n ; joint CM action.
$p_y^m(i, j; z, u)$	Missile position transition probability (Y_n^m), given target.
$p_z^m(i, j; y, u)$	Missile target transition probability (Z_n^m), given position.
$X_n \in \mathbf{S}$	Joint missile threat state at time n , $(Y_n^1, Z_n^1, \dots, Y_n^M, Z_n^M)$.
$p(i, j; u)$	Transition probability of joint threat X_n , given joint CM u .
$\mu^l(i, u) \in \Delta(\mathbf{U}^l(i))$	Randomized CM deployment policy of Platform l .
$\mu(i, u)$	Joint randomized CM deployment policy.
$g^l(i, u)$	Cost to platform l when $X_n = i$ and $U_n = u$.
$\mathbf{v}^l(i, \mu)$	Total expected cost starting from state i under policy μ .

Glossary of terms for Chapter 2.

Bibliography

- [62] D. Alberts, J. Garstka, and F. Stein. *Network centric warfare: developing and leveraging information superiority*. CCRP Publications, second edition, 1999.
- [63] E. Altman. *Constrained Markov decision processes*. Chapman and Hall/CRC, 1999.
- [64] D. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1 and 2. Athena Scientific, second edition, 2001.
- [65] S. Blackman and R. Popoli. *Design and analysis of modern tracking systems*. Artech House, 1999.
- [66] J. Breakwell and A. Merz. Football as a differential game. *Journal of Guidance, Control, and Dynamics*, 1992.
- [67] M. Breton. Algorithms for stochastic games. *In Stochastic games and related topics, Series C; Game theory, mathematical programming and operations research*, 1991.
- [68] L. Van Brunt, editor. *The glossary of electronic warfare*. EW Engineering, 1984.
- [69] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley, 1991.
- [70] J. Cruz and M. Simann et. al. Game-theoretic modeling and control of a military air operation. *IEEE Transactions on Aerospace and Electronic Systems*, 37(4), 2001.
- [71] H. Eustace, editor. *The International Countermeasures Handbook*. EW Communications, sixth edition, 1980.
- [72] J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer-Verlag, 1997.
- [73] S. Gutman. On optimal guidance for homing missiles. *Journal of Guidance and Control*, 1979.
- [74] P. Herings and R. Peeters. Stationary equilibria in stochastic games: structure, selection and computation. *METEOR Working Paper No. RM/00/31.*, 2000.

- [75] F. Thuijsman J. Filar, T. Schultz and O. Vrieze. Nonlinear programming and stationary equilibria in stochastic games. *Mathematical Programming*, 50, 1991.
- [76] Y. Lipman and J. Shinar. Mixed-strategy guidance in future ship defense. *Journal of Guidance, Control, and Dynamics*, 1996.
- [77] W.S. Lovejoy. Computationally feasible bounds for partially observed markov decision processes. *Operations Research*, 39(1), 1991.
- [78] W.S. Lovejoy. A survey of algorithmic methods for partially observed markov decision processes. *Annals of Operations Research*, 28(1), 1991.
- [79] M. Maskery and V. Krishnamurthy. Game-theoretic missile deflection in network centric warfare. *Proceedings of the IEEE International Conference on Networking, Sensing and Control*, 2005.
- [80] Joint Chiefs of Staff, editor. *Dictionary of Military and Associated Terms*. Department of Defense, 1987.
- [81] M. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, 1994.
- [82] S. Ross. *Introduction to probability models*. Academic Press, sixth edition, 1997.
- [83] A. Sen and D. Shanno. Computing nash equilibria for stochastic games. *Working Paper*, 2004.
- [84] L. Shapley. Stochastic games. *Proceedings of the National Academy of Sciences USA*, 39(10), 1953.
- [85] H. Sherali, Y. Lee, and D. Boyer. Scheduling target illuminators in naval battle-group anti-air warfare. *Naval Research Logistics*, 42, 1995.

Chapter 3

Decentralized Force Protection Against Anti-Ship Missiles: Correlated Equilibrium Approach ³

3.1 Introduction

Steady advances in communications technology, combined with advances in the theory of modern warfare, have combined to make high-bandwidth communications equipment widely available aboard military platforms. This equipment is highly advanced, capable of automatically and seamlessly synchronizing observations and actions among a large number of decision makers. The resulting potential for improved mission performance presents many challenges and opportunities, and the need for theoretical models for such endeavors is clear. Such models must consider the use of secure, wireless data networks to allow modern platforms to assist each other in the implementation of time-critical tasks, while respecting all the various goals and objectives present among members of the collective.

Nowhere is this situation more clear and immediate than in the problem of cooperatively and dynamically synchronizing countermeasures against anti-ship missiles among members of a Naval task group. In this scenario, several ships, communicating over a common data network, face an attack by a number of radar guided missiles, and must deploy electronic countermeasures (CM) to deflect or confuse the missiles, in an attempt to defend both themselves and other members of the task group. This situation is characterized by the need to make rapid, decentralized decisions guided by real-time communication, that simultaneously balance the requirements for self and group preservation.

In this chapter, we consider such a scenario, with missile threats modeled as a discrete Markov process. Missiles appear randomly on the perimeter of a fixed physical space, and

³A version of this chapter has been accepted for publication. Maskery, M. and Krishnamurthy, V. (2007) Network-Enabled Missile Deflection: Games and Correlated Equilibrium, IEEE Transactions on Aerospace and Electronics Systems 43:3.

move toward the ships under known targeting and guidance laws. Electronic countermeasure (CM) capabilities, which are either decoys or electromagnetic jamming signals, are modeled as ship action that are deployed dynamically to reduce the threat. Each platform represents a decision maker who observes the missile positions and uses local CMs to maximize own safety, while coordinating with other platforms doing the same.

The problem is formulated as a transient stochastic game with cost-minimizing players. The fact that players coordinate their actions through a network is reflected by considering the relevant solution to be a correlated equilibrium [88], [89], allowing players to coordinate actions in real time through some player-defined communication scheme, such that global and local objectives are simultaneously optimized. Such a strategy is referred to as a NEOps equilibrium strategy. The use of a correlated equilibrium is the major innovation of this chapter, distinguishing it from the results of Chapter 2. In that paper, players deployed countermeasures to deflect anti-ship missiles, but did not coordinate. The two approaches are complementary; Chapter 2 provides a highly robust and efficient missile deflection solution, since coordination is not required, while the present chapter accounts for the possibility of coordination as it arises in the NEOps framework.

Such scenarios arise out of the general framework of Network Centric Warfare (NCW) [86], and similar terms such as Network Enabled Operations, which denote a perspective on military operations emphasizing communication and coordination between friendly forces, in order to gain and leverage information superiority over an enemy. NCW connects independent, intelligent decision makers, allowing them to make joint decisions that are competitively optimal for each interested party. The game theoretic approach, as outlined here, is able to model such a situation, allowing us to define the types of strategies that would be acceptable in a NCW setting through the correlated equilibrium concept.

Examples of previous and current technologies enabling NEOps in a naval setting are the Cooperative Engagement Capability, the Link 16 data network and the Single Integrated Air Picture (SIAP). Such technologies fuse and circulate common sensor data to all decision makers in a networked force, providing platforms with the capability to coordinate joint actions in real time. It is important to note that the results in this chapter are not dependent on any specific technologies, but there is a requirement for some enabling technology capable of synchronizing tactical data and coordinating efforts in real time. The technologies cited above are evidence that such a requirement is feasible.

The correlated equilibrium of a stochastic game describes a solution for playing a dynamic game in which players are able to communicate but are self-interested. This is a key point to bear in mind; we view a correlated equilibrium as a descriptive object, illuminating

the properties of the set of joint strategies inevitably reached by such decision makers. We take this view throughout the chapter, requiring that players' joint strategies take the form of a correlated equilibrium without always being compelled to specify the communication structure underlying such a strategy. Particular attention will be paid to the NEOps equilibrium strategy, as such an object describes how well a group of players can perform from a global perspective.

In addition to the motivation developed above, recent work in decentralized machine learning has identified the set of correlated equilibrium as the limit set of many decentralized machine learning algorithms [107], [108], [97]. This learning approach, based on regret matching, requires minimal awareness from each player, and can hence be effectively employed in a decentralized setting. Players repeatedly simulate the game together and learn, in a completely decentralized fashion, a correlated equilibrium strategy from only their own observed costs and empirical distribution of past play. The existence of such a simple, convergent, distributed learning solution makes correlated equilibria worthy of study in their own right.

Communication in Network Enabled Operations is not unlimited, and a cooperative NEOps equilibrium strategy can be implemented only if channel capacity constraints can be satisfied. One of the key contributions of this chapter is to quantify the amount of communication necessary in the implementation of a NEOps equilibrium strategy for missile deflection. This is done through elementary concepts in information theory. As a consequence, we show how NEOps equilibrium strategies can be computed in the presence of bandwidth constraints, and provide examples showing the trade-off between performance and communication.

Main Results

The main contributions of this chapter are listed as follows.

- The formulation of a decentralized missile deflection problem as a transient stochastic game with communication. In Sections 3.2 and 3.3, we consider a scenario with ships as self-interested decision makers able to coordinate countermeasure deployment against a fixed number of missiles with known dynamics in real time.
- A theoretical foundation for Network Enabled Operations via the definition of NEOps equilibrium strategy. Sections 3.1.1 and 3.1.2 define and relate the concepts of Network Enabled Operations and correlated equilibrium. The NEOps equilibrium strategy is defined as the relevant competitive equilibrium strategy for Network Centric Warfare,

formulated as a stationary correlated equilibrium for the stochastic missile deflection game. The strategy is characterized as the solution to an optimization problem with bilinear constraints in Section 3.4.

- An information theoretic analysis of the communication overhead required for implementation of the NEOps equilibrium strategy. Communication overhead is shown to be related to the mutual information between the random variables describing player actions in Section 3.5. We also present several implementation architectures in that section.
- Numerical examples illustrating the NEOps equilibrium strategy for a moderate-scale missile deflection games are presented, and the trade-off between communication and performance is examined in Section 3.6.

3.1.1 Network Enabled Operations

Network Enabled Operations (NEOps) is an adaptation of the Network Centric Warfare concept, as put forth in [86]. In contrast with NCW, NEOps emphasizes the inclusion of operations other than war, and is not so explicitly bound to network technology. A candidate definition for Network Enabled Operations is given as follows [91]:

Definition 3.1.1 *Network Enabled Operations (NEOps) represent an approach to the conduct of military operations characterized by common intent, decentralized empowerment, and shared information, enabled by appropriate culture, technology and practices.*

A representative overview of the proposed operation of NEOps with regards to missile deflection is depicted in Figure 3.1. The enabling technology cited in Definition 3.1.1 is present in the form of a secure communication network capable of distributing sensor data and coordinating actions among a large number of military platforms in real time; each decision maker is provided with an identical, high quality picture of the battlespace by sharing sensor, decision, and engagement data between all interested parties.

We claim that our work supports Definition 3.1.1 in the following manner:

- Common Intent: The NEOps equilibrium strategy is optimal with respect to a single, collective objective, representing the common intent of the networked force.
- Decentralized Empowerment: The equilibrium aspect of the NEOps equilibrium strategy empowers each decision maker since it restricts attention to the set of strategies which are independently optimal from each individual point of view. Implicit in this

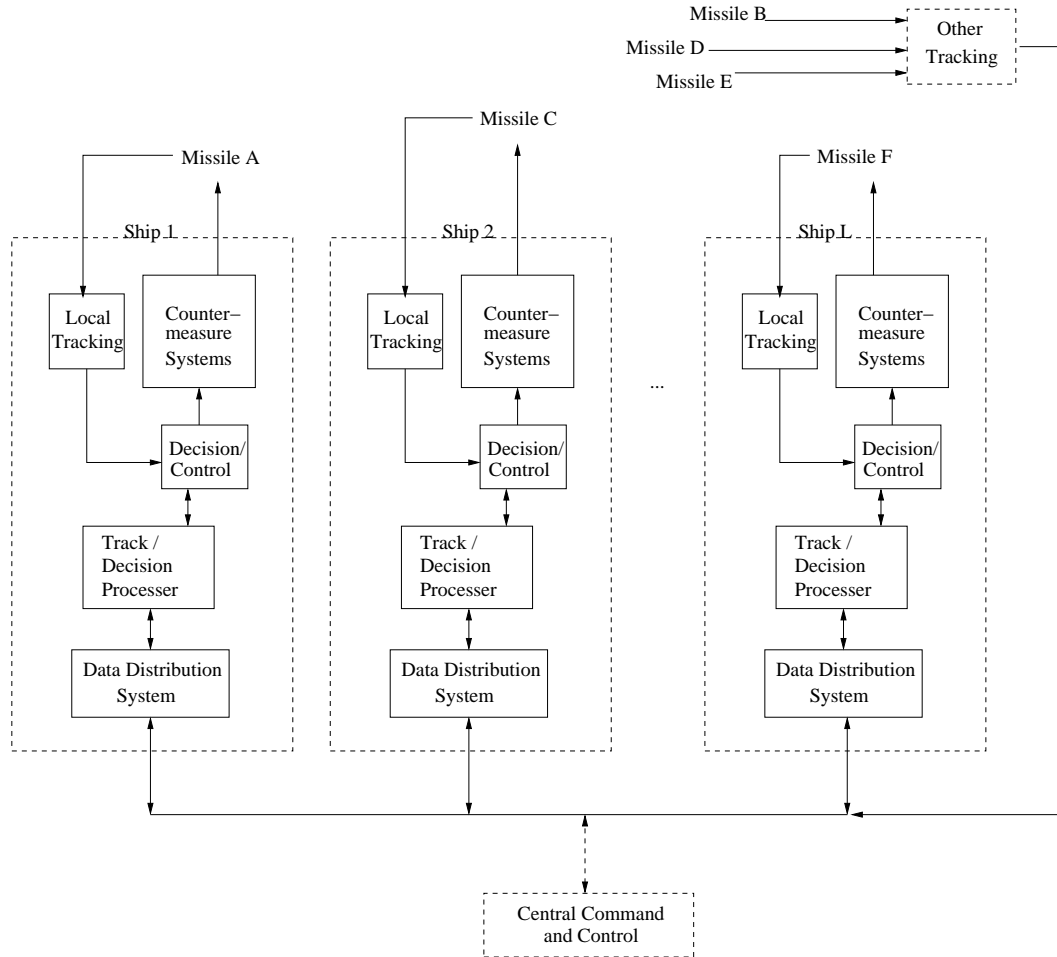


Figure 3.1: Decentralized missile deflection using Network Enabled Operations. Ships share sensor measurements and coordinate decisions through the network, with the possible assistance of a central command and control that issues acceptable recommendations.

interpretation is the existence of disparate individual objectives in addition to the common objective mentioned above. Furthermore, decentralized empowerment implies multiple, independent decision makers, justifying the use of game theory.

- **Shared Information:** This is supported in two ways. First, we assume that each decision maker shares common observations of the missile threat through the use of a common sensor network. Second, we allow users to share information about their reactions to these threats in real time. In game-theoretic terms, the first assumption is characterized by the fact that we do not use the theory of Bayesian games, as would be required if decision makers had disparate tactical information. The second assumption is characterized by use of the correlated as opposed to the Nash equilibrium solution concept.

3.1.2 The Case for Correlated Equilibrium in NEOps

The correlated equilibrium concept, introduced by Aumann in [88] and [89], is a generalization of the Nash equilibrium. For static noncooperative games, it describes the joint strategy of players who can communicate their intentions to each other (possibly through an intermediary), but are self-interested and are able to update conditional probabilities using Bayes' rule. Aumann showed that players can achieve a global performance improvement by implementing a correlated instead of a Nash equilibrium. The correlated equilibrium requires that actions be coordinated, but guarantees that each player retains an element of choice.

In its canonical form, a static correlated equilibrium consists of a game augmented by a correlation device, which sends “private recommendations” to each player. The players, knowing the equilibrium structure, find it rational to follow these recommendations because of the *a posteriori* action probabilities they imply about other players [88]. A Nash equilibrium is trivially a correlated equilibrium because it is a product distribution; players receiving recommendations from a Nash distribution learn nothing new about others' actions, but find it rational to follow the recommendation because of the classical “best response” rationale.

Although it is a useful tool, both in theory and practice, we do not justify the use of correlated equilibrium in this chapter based on the use of such an explicit correlation device. Instead, we reason, as in [89], that all policies adopted by independent, rational decision makers with the ability to coordinate will take the form of a correlated equilibrium. We then characterize the NEOps equilibrium strategy as a “globally optimal” correlated equilibrium,

and examine the communication required for its implementation. A correlated equilibrium, in this last sense, is defined as follows.

Definition 3.1.2 *Let Ω be a finite information set, and define partitions \mathcal{P}^l on Ω for each game player $l \in L$. Define player strategies $\mu^l(\omega) : \omega \in \Omega$, such that $\mu^l(\omega) = \mu^l(\omega')$ whenever ω and ω' are elements of $P^l \in \mathcal{P}^l$. Also define a probability measure p on Ω . A correlated equilibrium is a set of strategies $\{\mu^l : l \in L\}$ such that, for every τ^l such that $\tau^l(\omega) = \tau^l(\omega')$ whenever ω and ω' are in $P^l \in \mathcal{P}^l$,*

$$\sum_{\omega \in \Omega} p(\omega) u^l(\mu^l(\omega), \mu^{-l}(\omega)) \leq \sum_{\omega \in \Omega} p(\omega) u^l(\tau^l(\omega), \mu^{-l}(\omega)). \quad (3.1)$$

Here, u^l represents the cost to Player l , and μ^{-l} represents the collection of strategies of all players but l .

At each decision point of the game, each player l receives partial information about ω through the observation $\omega \in P$ for some $P \in \mathcal{P}^l$. This information represents the knowledge about the other players' decisions, along with any other relevant information, which is factored into Player l 's action decision via (3.1). Elements of Ω describe the exact value of all parameters as to which a player may be uncertain, including the exact actions taken by each player in the game. To simplify matters, we assume in this chapter that this uncertainty in others' actions is the only source present. That is, we restrict our attention to the sets $\Omega_s = \times_{l \in L} U^l(s)$, where L is the set of players, s is a known parameter, and $U^l(s)$ is the set of actions available to player l given parameter s .

Example If each $\mathcal{P}^l = \{\Omega, \emptyset\}$, then players do not communicate. In this case, the correlated equilibrium is equivalent to a Nash equilibrium. If each \mathcal{P}^l is identical, the players have fully public knowledge of each others' actions.

It is well known that this last case, corresponding to public communication, is not always optimal. Players can improve their overall performance by restricting themselves to complex private communication schemes. For this reason, we allow the information partition to be a free variable in our formulation, so that we may later analyze its structure under optimal conditions. We do, however, pay special attention to the public communication case, as it is particularly easy to implement.

It is useful to make two remarks here, due to Aumann [89]. First, although a correlated equilibrium generally takes the form of a probability distribution over joint actions, its

probabilistic nature reflects only the uncertainty of players about the action choices of others; players themselves do not explicitly randomize their actions. This is an important observation for those who would be uncomfortable entrusting outcomes to a machine that essentially performs coin flips. Second, every correlated equilibrium corresponds to at least one information system. That is, we are justified in characterizing the correlated equilibrium solution *before* specifying the information structure that enables such an equilibrium.

Although this view of a correlated equilibrium is philosophically sound, we acknowledge that there is a gap in current understanding between Aumann’s *descriptive* view of correlated equilibrium, and the many *prescriptive* means of implementing it. The NEOps equilibrium strategy can be motivated as a reasonable description of actions, but the implementation method must remain unspecified. Conversely, we may specify an implementation method (see Section 3.5.2), but this affects the claim that the strategy has been freely negotiated. We solve this problem by supporting both the descriptive and prescriptive views separately, with the understanding that in practice either one view or the other will be useful. The information theoretic portion of the chapter is valid from both perspectives.

The extension of the correlated equilibrium to stochastic games originated with [103], in which the prototype concept of extensive form correlated equilibrium was defined. The natural progression to stochastic games is apparent in papers such as [113], which defines a stationary correlated equilibrium for a stochastic game, but considers only the special case of symmetric information, and [106], which uses correlated equilibria as the goal of a reinforcement learning procedure for adaptive game-theoretic agents. More recently, [119] shows the existence of a (nonstationary) correlated equilibrium based on retaliation strategies.

In contrast to the above examples, this chapter focuses on a stochastic correlated equilibrium that is both stationary and allows for asymmetric information transfer between players. Such a definition is advantageous in that it avoids the performance limitations associated with symmetric information, as is the case in [113], and is not vulnerable to the perpetual performance degradation that may result from retaliation to errors in play in a nonstationary equilibrium of the type described in [119]. On the other hand, we restrict our attention to stochastic game models with finite state and action spaces. The main reason for this, besides ease of exposition, is that the stochastic version of the correlated equilibrium solution concept is only known to exist, in general, for this case. Notwithstanding such existence problems, extension of the concepts presented here to countable and continuous spaces is relatively straightforward.

For purposes of analysis, the calculation of an optimal correlated equilibrium for a stochastic game requires the solution of an optimization problem with linear objective and

bilinear constraints. In order to compute a solution to such a problem using standard non-linear programming methods, an initial feasible solution is needed. Since a Nash equilibrium is also a correlated equilibrium, it will be convenient to use a Nash equilibrium as a starting point for such a calculation. This is true since Nash equilibria can often be computed using decentralized, iterative techniques such as a best response scheme as in Chapter 2. Such techniques are of relatively low dimension and are based on an intuitive approach, giving a simple starting point for correlated equilibrium calculations.

In this light, one may regard a NEOps equilibrium strategy as an improvement over a Nash equilibrium resulting from the sudden discovery of coordination abilities; players start off by playing a Nash equilibrium, then gradually improve their situation through communication by jointly computing better and better correlated equilibria, driven by some mutually agreed upon global objective. The optimal correlated equilibrium is a steady state of such an improvement process. In a machine learning context, such a steady state might arise through fictitious play, using methods such as the regret matching schemes discussed in the introduction.

Our approach is game-theoretic in nature, which is meant to reflect the decentralized nature of command and facilitate distributed computation. However, other approaches are possible. A comparable approach is found in [118], which considers a target illumination problem very similar to our multiple-ship, multiple-missile deflection scenario. This chapter computes an optimal scheduling solution for defeating the missile threat, which is implemented through full coordination. The optimal scheduling depends heavily on centralized control, but appears quite effective overall. In addition, one can formulate a max-min fair approach [114] in which the objective is to minimize the maximum probability of hit over all players. However, the computational complexity involved is quite high, and the solution again does not reflect decentralized empowerment. Finally, one can formulate a constrained Markov decision problem [87] which directly minimizes the probability of hit, subject to the probability of hit of each player being below a given threshold. This approach has the advantage that only a linear program, as opposed to our bilinear program, must be solved. However, it represents a weaker form of decentralized empowerment, since players may only specify a lower performance bound, instead of insisting on the best available performance. Moreover, it is not clear how the thresholds should be chosen, so an outer loop would also be needed.

3.2 Network Enabled Operations in Dynamic Missile Deflection

In this section, we describe our application domain, as originally formulated in Chapter 2, in a manner amenable to NEOps. We consider a missile threat evolving according to a discrete Markov process with dynamics jointly controlled by the countermeasure (CM) decisions of each ship in a task group. Fundamental to this model are the following:

- A set of incoming missiles M .
- A set of CM-equipped defensive agents L .
- A finite state space \mathbf{S} , describing possible states of the multiple missile threat.
- A set of countermeasures $U^l(i)$ available to each agent $l \in L$, for each state $i \in \mathbf{S}$.
- A discrete-time Markov process describing the evolution of the aggregate missile threat on \mathbf{S} .

We slightly abuse notation by allowing M and L to represent both a set and its cardinality.

We assume that the theater of operations is a discrete space, and that missiles move in discrete time toward the task group. We also consider the task group to be stationary, due to the relative speed difference between missiles and ships, the short engagement time, and the fact that ship maneuvers may be restricted. A group of autonomous players defends the task group, each player having control over a subset of all countermeasures and charged with defending a given subset of ships. Given the aggregate missile state (positions, directions, etc.) at any time index, the task of each player is to deploy a sequence of CM so as to defend their charges, without exact knowledge of the CM decisions being made simultaneously by the other players, who in turn are carrying out their similar tasks.

A key contribution of Network Enabled Operations is to provide common, high quality track information to all decision makers. Hence we make the assumption that all decision makers have complete observability over the threat state at all times. In practice, the missile process will be only partially observable, due to the presence of noisy measurements, uncertainty of missile type, etc. In this case, our assumption of complete observability is equivalent to the use of certainty equivalent control [94]. At any rate, the optimal control results of this chapter must first be understood before tackling the partially observed case.

The remainder of this section models the missile threat as a Markov process. The formulation of a single missile threat is considered first, which is then extended to multiple missile threats.

3.2.1 Single Missile Process

We begin with the evolution of a single missile, labelled missile m , indexed by discrete time t . The missile moves through a physical battlespace, guided by radar towards a given target that evolves with time. The state is given by

$$\begin{pmatrix} X_t^m \\ Z_t^m \end{pmatrix} = \begin{pmatrix} \text{Position of missile } m \text{ at time } t \\ \text{Target of missile } m \text{ at time } t \end{pmatrix}. \quad (3.2)$$

Define the battlespace B to be the set of all possible positions of an active missile. We take this to be a two-dimensional grid, which is sufficient for sea-skimming missiles. For example a 600×600 grid with each coordinate representing a missile's physical position to within 50 metres is sufficient to reflect the position of a missile with a range of 30 kilometres, a speed of 500 metres per second, and accurate position updates made 10 times per second.

Define $T \subset B$ to be the set of all possible targets of a missile. We assume that a missile moves more or less directly from its current position towards its current target, but that this target may change according to the missile's targeting decisions (which may be influenced through ship countermeasures). Since missiles can be misled, not every element of T corresponds to a ship location, and we define $\mathfrak{T} \subset T$ to be the set of true target positions occupied by the task group.

An active missile has a physical position X_t^m in B and a current target Z_t^m in T . To this we add two special states, a pre-engagement state r , corresponding to missile m being a potential but as-yet untracked threat, and a terminal state Δ , corresponding to a missile has reached a target, terminated its flight, or has acquired a trajectory that ensures it is no longer a threat. When one of these events occurs, the missile transitions to the unique Δ with probability one. The state space of the missile is denoted by

$$\mathbf{S}^m = \{B \times T, \{r\}, \{\Delta\}\}. \quad (3.3)$$

For states $s \in B \times T$, we say $s = (s_x, s_z)$, where s_x and s_z denote position and target components, respectively.

In this chapter, (X_t^m, Z_t^m) is assumed to be a Markov process, controlled in part by CM actions, with transition probabilities defined as follows.

Definition 3.2.1 *Let u_t represent some joint CM action of all agents at time t . For $i, j \in$*

$B \times T$, the physical and targeting transition probabilities of missile m are given by

$$p_x^m(i, j; z, u) = \Pr(X_{t+1}^m = j | X_t^m = i, Z_t^m = z, u_t = u), \quad (3.4)$$

$$p_z^m(i, j; x, u) = \Pr(Z_{t+1}^m = j | Z_t^m = i, X_t^m = x, u_t = u). \quad (3.5)$$

These probabilities are assumed to be conditionally independent, hence the overall transition probabilities are given by

$$p^m(i, j; u) = \Pr((X_{t+1}^m, Z_{t+1}^m) = (j_x, j_z) | (X_t^m, Z_t^m) = (i_x, i_z), u_t = u) \quad (3.6)$$

$$= p_x^m(i_x, j_x; i_z, u) p_z^m(i_z, j_z; i_x, u). \quad (3.7)$$

Furthermore, we define the following transition probabilities:

$$p^m(i, \Delta; u) = 1 \text{ whenever } i \in \mathfrak{T} \times T, \quad (3.8)$$

(missile terminates upon reaching a target)

$$p^m(\Delta, \Delta; u) = 1, \quad (3.9)$$

(terminated missiles do not recover)

$$p^m(r, j; u) = \beta^m(j), j \in S^m, \quad (3.10)$$

(activation probabilities are independent of control),

where β^m is a given probability distribution.

We account for two sources of randomness in the missile process. That is, randomness in the physical guidance system, and randomness in the targeting mechanism. Other sources, such as turbulence and errors in the sensor network's estimates of the missile position can be included as desired.

For the physical guidance system, we assume the missile operates according to proportional guidance as described in [95]. That is, the missile moves on an intercept course towards a target by predicting its position at the time of impact. However, because the missile may only make noisy position and velocity estimates of its target, occasional deviations from the proper course may occur. For example, suppose a missile is at point $(X_t^m, Z_t^m) = (a, z)$ a time t . Then we can and is on a straight-line intercept course that should take it next to point $b(a, z)$ on its way to target position z . We may account for random guidance errors by assuming that the missile moves to $b(a, z)$ with some high probability q , and otherwise moves to one of the two states adjacent to both a and $b(a, z)$ with probability $(1 - q)/2$.

The targeting transition probabilities depend on some programmed decision rules of the missile, but targets can also “drift” if the missile is viewed as a receiver operating in a noisy environment. Suppose that $Z_t^m = a$, and its radar receives information about the target position over a discrete memoryless channel [99]. The channel randomly reports the new position as a' with probability $p_z^m(a, a'; x, u)$ and the missile updates Z_{t+1}^m accordingly. The distortion may or may not vary with the missile position. An effective countermeasure u might minimize $p_z^m(a, a; x, u)$ when $a \in \mathfrak{T}$, and maximize it otherwise.

3.2.2 Multiple Missile Process

Naval task groups must be prepared not only for isolated missile attacks, but also for the possibility of large-scale, multiple missile attacks. In this section we extend the Markov missile process to the multiple M missile case, which evolves on a much larger, product state space.

Define the aggregate missile trajectory indexed over time t by

$$Y_t = (X_t^1, Z_t^1, X_t^2, Z_t^2, \dots, X_t^M, Z_t^M). \quad (3.11)$$

The process evolves on the product state space \mathbf{S} , defined as

$$\mathbf{S} = \mathbf{S}^1 \times \mathbf{S}^2 \times \dots \times \mathbf{S}^M. \quad (3.12)$$

Since there is a unique absorbing state Δ for each missile, it is clear that $(\Delta, \Delta, \dots, \Delta) \in \mathbf{S}$ is the unique absorbing state for the aggregate process. For simplicity, we refer to this state simply as Δ ; whether this refers to a single or aggregate terminal state will be clear from the context. We also define \mathbf{S}' to represent all states in \mathbf{S} except Δ . A generic state $s \in \mathbf{S}$ is written as (s^1, s^2, \dots, s^M) .

We assume that each missile evolves independently (they do not communicate). In this case the transition probabilities of the Y_t process may be written as

$$p(i, j; u) = \prod_{k=1}^M p^k(i^k, j^k; u), \quad (3.13)$$

All the dynamics of the single missile process carry over to the multiple missile case.

This formulation allows us to consider a variety of threat situations. Not only does each distinct value of M represent a fundamentally different threat situation, but the potential for mixtures of different missile types leads to a combinatorially large number of situations

for each M . In practice, only a small number of threat situations would be considered, chosen to be representative of the types of situations anticipated.

Countermeasures (CM) involve mechanisms such as physical decoys, chaff, and electromagnetic jamming [100]. The primary function of each CM type is to confuse the targeting system of the missile, possibly by providing a more attractive (and false) target. It is assumed that CM control decisions can be made dynamically, that is, a control decision is made at each time index. This is a reasonable assumption for unconstrained countermeasures such as electromagnetic jamming. It is also straightforward to adjust the problem so as to consider more limited countermeasures, without affecting the theory here, as in Chapter 2.

Although we refrain from going into detail, we assume that each joint CM action u affects the missile process by controlling the transition probabilities $p(i, j; u)$ in a known way. This approach allows one to specify varying CM effects for different missile states, and explicitly characterize the uncertainty associated with each countermeasure attempt. For more details, and an extended model of the missile dynamics, with multiple missiles, decoys, noisy missile targeting, and CM constraints, the reader is referred to Chapter 2 and [111].

3.3 Decentralized Missile Deflection as a Transient Stochastic Game

Transient stochastic games [117],[101] consist of a Markov process evolving on a state space \mathbf{S} , along with a set L of players, who influence the transition probabilities of the process by choosing one of a finite number of actions in each state. Each player associates a cost with each state-action combination, and chooses actions with the objective of minimizing the total expected cost accumulated over the lifetime of the process. The transience property is satisfied by having a cost-free, absorbing state $\Delta \in \mathbf{S}$, such that the probability of reaching Δ in a finite number of steps from any state $i \in \mathbf{S}$ is nonzero regardless of actions. It is easy to see that the missile deflection problem is a transient stochastic game (with CM platforms taking the role of players), since a missile will reach its target, fly out of range, or run out of fuel in finite time.

The instantaneous costs incurred by player $l \in L$ when the missile is in state i and CM action u is taken is defined as $g^l(i, u)$. Since state Δ is cost free, $g^l(\Delta, u) = 0$ for all $l \in L$, and all action choices u . A particularly simple and appropriate cost function for the missile

deflection problem is obtained as

$$g^l(i, u) = I(i_x^k \in \mathfrak{T}^l, \text{ some } k). \quad (3.14)$$

It can be shown, using (3.8), that the total expected cost under (3.14) is equivalent to the expected number of missiles that ever reach \mathfrak{T}^l , Chapter 2, [116],[111]. In the case of a single threat, it is also the probability of a missile hit. (The cost function can be defined in other ways, for example to penalize a player whenever the missile reaches a new minimum distance, [111].) Although (3.14) is independent of u , we maintain generality by including the dependence of g^l on u below.

In this chapter, we consider the game solution to be a stationary correlated equilibrium from the point of view of an outside observer. From such a perspective, players select joint CM actions according to a stationary probability distribution. This probability distribution is referred to here as either a stationary randomized joint strategy. Let \mathbf{U}^l (resp. \mathbf{U}) represent the vector of available individual (resp. joint) action sets over all states i . That is,

$$\mathbf{U}^l = (U^l(1)^T, U^l(2)^T, \dots, U^l(|\mathbf{S}|)^T)^T, \quad (3.15)$$

$$\mathbf{U} = (U(1)^T, U(2)^T, \dots, U(|\mathbf{S}|)^T)^T, \quad (3.16)$$

(where superscript T indicates vector transpose). Furthermore, let $\Delta(\mathbf{U}^l)$ (resp. $\Delta(\mathbf{U})$) represent the set of probability distributions over the rows of \mathbf{U}^l (resp. \mathbf{U}). Then we are led to the following definition.

Definition 3.3.1 *A stationary randomized joint strategy μ is mapping from \mathbf{S} to $\Delta(\mathbf{U})$. That is, for each state $i \in \mathbf{S}$, $\mu_i(\cdot)$ is a probability distribution over actions in $U(i)$. The probability of players jointly selecting action u in state i is $\mu_i(u)$.*

Each joint strategy μ induces a Markov chain on \mathbf{S} , with transition probabilities,

$$p(i, j; \mu) = \sum_{u \in U(i)} p(i, j; u) \mu_i(u), \quad (3.17)$$

and expected costs,

$$g^l(i, \mu) = \sum_{u \in U(i)} g^l(i, u) \mu_i(u), \quad (3.18)$$

for each player $l \in L$. Given a joint strategy, players associate a total expected cost $G^l(i, \mu)$

with each state, defined through the (recursive) set of equations

$$G^l(i, \mu) = g^l(i, \mu) + \sum_{j \in \mathbf{S}'} p(i, j; \mu) G^l(j, \mu). \quad (3.19)$$

This dynamic programming equation is well-known for stochastic games, [117], [101], [94]. The reader can easily verify that it is finite for all transient stochastic games.

The solution to a stochastic game is an equilibrium; a joint strategy which is simultaneously optimal for all players. The study of equilibria in stochastic games is usually performed by considering a set of auxiliary strategic games [117],[101], defined as follows. Consider a single state i , and assume that some strategy μ will be followed in every state following i . Then each player could, in principle, predict the expected total costs associated with every joint action u in state i . This is done by adding the current cost of action u with the expected cost incurred from the next state onward, assuming strategy μ . These costs define an auxiliary game for i , with costs

$$G^l(i, (u, \mu_{-i})) = g^l(i, u) + \sum_{j \in \mathbf{S}'} p(i, j; u) G^l(j, \mu), \quad (3.20)$$

where (u, μ_{-i}) corresponds to the strategy in which joint action u is taken immediately in state i , and strategy μ is used in every subsequent state, including any return to i . A stationary joint strategy μ that is simultaneously a Nash equilibrium for all auxiliary games is a Nash equilibrium for the stochastic game [101]. (That is, each player strategy is a best response in terms of present and future payoffs in every state.)

Similarly, a stationary joint strategy that is simultaneously a correlated equilibrium for all auxiliary games will be a correlated equilibrium for the stochastic game.

3.4 Characterization of the NEOps Equilibrium Strategy

In this section we formulate the NEOps equilibrium strategy as the solution to a constrained optimization problem, in which the constraints themselves are dependent on the solution. The correlated equilibrium we characterize is optimal with respect to some common linear objective function, subject to individual “equilibrium” constraints, and so characterizes the strategy form of ships connected through NEOps. This is in contrast to Chapter 2, where communication is unavailable and a Nash equilibrium strategy is the best achievable.

The NEOps equilibrium strategy can be computed as the solution to a Markov decision problem [87], in which the search space is confined to policies that are correlated equilibria.

Similarly to the player costs (3.14), we may take the collective costs as

$$g(i, u) = I(i_x^k \in \mathfrak{T}, \text{ some } k). \quad (3.21)$$

Here, $\mathfrak{T} = \cup_{l \in L} \mathfrak{T}^l$, where each \mathfrak{T}^l represents the set of target locations that player l is charged with defending. The collective objective is thus to minimize the expected number of hits to *any* ship.

The linear programming approach to Markov decision problems found in [87] allows us to compute the NEOps equilibrium strategy via an optimization. The idea is that we may formulate the objective and policy in terms of an *occupation measure* π , which itself represents the decision variables in an optimization problem. We give an intuitive outline of the procedure below; the reader may refer to [87], [115] and [94] for a more rigorous treatment.

Define $\pi(i, u)$ to be the expected number of times state i is reached and joint CM u is deployed. Then clearly the expected total cost incurred during an attack satisfies

$$E[\text{total cost}] = \sum_{i \in \mathbf{S}'} \sum_{u \in U(i)} g(i, u) \pi(i, u). \quad (3.22)$$

Moreover, due to (3.21), the expected total cost is equal to the expected number of missile hits.

Assume that the initial distribution of the missile threat is uniform over \mathbf{S}' (the initial distribution, it turns out, is largely irrelevant to the optimal policy). It can be shown (Chapter 8 of [87]), due to the nature of Δ as an absorbing state, that the set of occupation measures π is exactly the set of π satisfying

$$\begin{aligned} \sum_{u \in U(i)} \pi(i, u) &= \sum_{j \in \mathbf{S}'} \sum_{u \in U(j)} p(j, i; u) \pi(j, u) + \frac{1}{|\mathbf{S}'|}, \\ \pi(i, u) &\geq 0, \end{aligned}$$

for all i, u . This can be viewed as a restriction to measures π that satisfy the balance equations of the underlying Markov chain. Moreover, due to Δ , we may implement any π via a stationary policy defined as follows:

Definition 3.4.1 *When the threat is in state i , choose joint CM u with probability*

$$\mu_i^{(\pi)}(u) = \frac{\pi(i, u)}{\sum_{v \in U(i)} \pi(i, v)}. \quad (3.23)$$

Determining the NEOps equilibrium strategy is thus equivalent to solving the following mathematical program:

$$\begin{aligned}
 & \text{minimize } \sum_{i \in \mathbf{S}'} \sum_{u \in U(i)} g(i, u) \pi(i, u) \\
 & \text{subject to } (\forall i \in \mathbf{S}', l \in L, u^l, d^l \in U^l(i)) : \\
 & \quad \sum_{u \in U(i)} \pi(i, u) = \sum_{j \in \mathbf{S}'} \sum_{u \in U(j)} p(j, i; u) \pi(j, u) + \frac{1}{|\mathbf{S}'|}, \\
 & \quad \mu^{(\pi)} \text{ is a correlated equilibrium.}
 \end{aligned} \tag{3.24}$$

We now formalize the last constraint of (3.24). In what follows we drop the superscript (π) for notational simplicity.

Definition 3.4.2 *A stationary randomized joint strategy μ is a stationary correlated equilibrium for a transient stochastic game if*

$$\sum_{u^{-l} \in U^{-l}(i)} \mu_i((u^l, u^{-l})) \left[G^l(i, ((u^l, u^{-l}), \mu_{-i})) - G^l(i, ((d^l, u^{-l}), \mu_{-i})) \right] \leq 0, \tag{3.25}$$

for all $i \in \mathbf{S}', l \in L, u^l, d^l \in U^l(i)$, where the $G^l(i, \cdot)$ are defined recursively through (3.19) and (3.20).

(Recall that the notation u^{-l} represents the actions all players but l .) The interpretation is that a correlated equilibrium is a probability distribution over joint actions which has the property that unilateral deviations are not profitable. (It is widely accepted that it is sufficient to consider one-shot deviations as above for stochastic games [109].)

This definition is a straightforward interpretation based on both (3.20) and (3.1). If, at each decision point and after communication with others, player l decides to take CM action u^l , then it must be because no other CM action d^l would yield a lower probability of hit to \mathfrak{T}^l . Implicit in this definition is that the NEOps equilibrium strategy μ constitutes the common prior probability in (3.1) and is thus common knowledge.

Substituting (3.23) into (3.25) yields the equivalent expression

$$\sum_{u^{-l} \in U^{-l}(i)} \pi(i, (u^l, u^{-l})) \left[G^l(i, ((u^l, u^{-l}), \mu_{-i})) - G^l(i, ((d^l, u^{-l}), \mu_{-i})) \right] \leq 0. \tag{3.26}$$

The NEOps equilibrium strategy is therefore given by the solution to the following mathe-

mational program:

$$\begin{aligned}
 & \text{minimize } \sum_{i \in \mathbf{S}'} \sum_{u \in U(i)} g(i, u) \pi(i, u) \\
 & \text{subject to } (\forall i \in \mathbf{S}', l \in L, u^l, d^l \in U^l(i)) : \\
 & \quad \sum_{u \in U(i)} \pi(i, u) = \sum_{j \in \mathbf{S}'} \sum_{u \in U(j)} p(j, i; u) \pi(j, u) + \frac{1}{|\mathbf{S}'|}, \\
 & \quad \sum_{u^{-l} \in U^{-l}(i)} \pi(i, (u^l, u^{-l})) \left[G^l(i, ((u^l, u^{-l}), \mu_{-i})) - G^l(i, ((d^l, u^{-l}), \mu_{-i})) \right] \leq 0, \\
 & \quad \pi(i, u) \geq 0 \quad \forall u \in U(i).
 \end{aligned} \tag{3.27}$$

Using (3.20), (3.26) can be rewritten as

$$\begin{aligned}
 & \sum_{u^{-l} \in U^{-l}(i)} \pi(i, (u^l, u^{-l})) \left[g^l(i, (u^l, u^{-l})) - g^l(i, (d^l, u^{-l})) \right] \leq \\
 & \sum_{u^{-l} \in U^{-l}(i)} \pi(i, (u^l, u^{-l})) \sum_{j \in \mathbf{S}'} \left[p(i, j; (d^l, u^{-l})) - p(i, j; (u^l, u^{-l})) \right] G^l(j, \mu).
 \end{aligned} \tag{3.28}$$

(3.28) clearly depicts the nature of the stationary correlated equilibrium. The left hand side represents the immediate benefit (in terms of reduced cost) of following the correlated equilibrium, while the right hand side represents the future benefit of deviating from it. The equation can be satisfied in three ways. If the right hand side is positive, then the proposed deviation is future profitable, but the present loss is too high to justify it. If the right hand side is negative and the left hand side positive, then the proposed deviation is profitable neither in the present nor the future. Finally, if the both sides are negative, then future losses are too high to justify an otherwise immediately profitable deviation.

We conclude this section by establishing the feasibility of the NEOps optimization problem (3.27). This requires the following result:

Lemma 3.4.1 *A stationary Nash equilibrium is a stationary correlated equilibrium.*

Proof Joint strategy μ is a stationary Nash equilibrium if and only if each μ_i is a Nash equilibrium for the auxiliary stage game associated with state i . But if μ_i is a Nash equilibrium, it is also a correlated equilibrium for the auxiliary stage game i , [89]. But this is precisely the definition of a stationary correlated equilibrium, so μ is also a stationary correlated equilibrium. ■

Lemma 3.4.1 immediately establishes the existence of stationary correlated equilibria. Since stationary Nash equilibria are known to exist for stochastic games of the type described here [101], stationary correlated equilibria must also exist. Thus, there is at least one feasible point of (3.27), so the problem is well-defined.

The global objective, defined through costs $g(i, u)$, defines the optimality criteria for the NEOps equilibrium strategy. In the missile deflection context, we have defined a strategy to be optimal if all ships are defended equally. However, optimality can be defined in any number of ways. Some examples (see [106] and reverse the role of cost and reward) are given below.

- Utilitarian: $g(i, u) = \sum_{l \in L} g^l(i, u)$.
- Republican: $g(i, u) = \min_{l \in L} g^l(i, u)$.
- Egalitarian: $g(i, u) = \max_{l \in L} g^l(i, u)$.

In this chapter, the global objective (3.21) can be regarded as either utilitarian or egalitarian. Differing player priorities, corresponding to different ship classes for example, can be incorporated into the definitions above by introducing appropriate weighting parameters.

3.4.1 Correlated Equilibrium as the Solution to a Bilinear Program

To solve (3.27), we must reformulate the problem to make the constraints independent of the solution. To this end, we introduce dummy decision variables $J^l(i)$, defined as the expected state-action costs satisfying (3.19). The optimization problem (3.27) can then be rewritten as

$$\begin{aligned}
 & \text{minimize } \sum_{i \in \mathbf{S}'} \sum_{u \in U(i)} g(i, u) \pi(i, u) \\
 & \text{subject to } (\forall i \in \mathbf{S}', l \in L, u^l, d^l \in U^l) : \\
 & \quad \sum_{u \in U(i)} \pi(i, u) = \sum_{j \in \mathbf{S}'} \sum_{u \in U(j)} p(j, i; u) \pi(j, u) + \frac{1}{|\mathbf{S}'|}, \\
 & \quad \sum_{z \in U(i)} \pi(i, z) J^l(i) = \sum_{u \in U(i)} g^l(i, u) \pi(i, u) + \sum_{j \in \mathbf{S}'} \sum_{u \in U(j)} p(i, j; u) \pi(i, u) J^l(j), \\
 & \quad \sum_{u^{-l} \in U^{-l}(i)} \pi(i, (u^l, u^{-l})) \left[g^l(i, (u^l, u^{-l})) - g^l(i, (d^l, u^{-l})) \right] \leq
 \end{aligned}$$

$$\sum_{u^{-l} \in U^{-l}(i)} \pi(i, (u^l, u^{-l})) \sum_{j \in \mathbf{S}'} \left[p(i, j; (d^l, u^{-l})) - p(i, j; (u^l, u^{-l})) \right] J^l(j),$$

$$\pi(i, u) \geq 0 \quad \forall u \in U(i). \quad (3.29)$$

(3.29) falls into the general class of quadratic programs, which have been studied in various forms. In fact, (3.29) has a more specific structure: it is an optimization problem with a linear objective and bilinear constraints.

Standard nonlinear solutions techniques for (3.29) involve starting at an initial feasible point and searching for improved feasible solutions. This initial feasible point may be taken, in some cases, to be a Nash equilibrium. Indeed, it has been observed that any Nash equilibrium of a static game lies on the boundary of the space of correlated equilibria [112]. This result can be straightforwardly extended to stochastic games, as in the following theorem.

Proposition 3.4.1 *In any finite stochastic game, the stationary Nash equilibria lie on the boundary of the space of stationary correlated equilibria.*

Proof A stationary Nash equilibrium consists of a Nash equilibrium for each auxiliary state game, that is, for the set of static games with player costs given by $G^l(i, (u, \mu_{-i}))$. According to Proposition 1 of [112], a Nash equilibrium either assigns zero probability to one or more joint actions, or renders every player indifferent among strategies. This implies that, for each $i \in \mathbf{S}$, either $\mu_i(u) = 0$ for some u , or (3.25) is satisfied with equality. Equivalently, for each i , one of the two inequality constraints in (3.29) must be satisfied with equality. Since the constraints of (3.29) describe the space of all correlated equilibria, with equality corresponding to boundaries, every stationary Nash equilibrium lies on the boundary of the space.

If one is able to compute an equilibrium in the absence of communication, one may use it as a starting point for computing an improved equilibrium with communication. Of course, a Nash equilibrium may in general be much more difficult to obtain than a correlated equilibrium, but we point out its usefulness here, since it has been the subject of much study, even in the decentralized missile deflection game in Chapter 2, [111].

(3.29) can be written in a more compact form,

$$\begin{aligned} & \text{minimize } g^T x \\ & \text{subject to} \end{aligned}$$

$$\begin{aligned}
Ax &= a, \\
x^T B_i y - x^T b_i &= 0, \quad i = 1, \dots, n_1, \\
x^T C_j y - x^T c_j &\geq 0, \quad j = 1, \dots, n_2, \\
x &\geq 0.
\end{aligned} \tag{3.30}$$

Decision variables π and J^l have been replaced by x and y , respectively. Because of the structure of the B_i , y is determined uniquely by x . We also note that $n_1 = |\mathbf{S}'| \times |L|$, and $n_2 = |\mathbf{S}'| \times \sum_{j \in L} (|U^j|)^2$, so $n_2 > n_1$.

We remark here that the NEOps equilibrium strategy is Pareto optimal among the class of correlated equilibrium strategies, when the global objective is utilitarian. Since the global objective g is (for any nonzero weighting) a sum of the individual objectives g^l , (3.38) corresponds to a scalarized version of the optimization problem:

$$\begin{aligned}
&\text{minimize } ((g^1)^T x, (g^2)^T x, \dots, (g^L)^T x) \\
&\text{subject to} \\
&Ax = a, \\
&x^T B_i y - x^T b_i = 0, \quad i = 1, \dots, n_1, \\
&x^T C_j y - x^T c_j \geq 0, \quad j = 1, \dots, n_2, \\
&x \geq 0.
\end{aligned} \tag{3.31}$$

It is well known that the solution to (3.30) is then a Pareto optimal solution to (3.31) [96].

In this chapter (3.30) is solved by standard nonlinear optimization (i.e. line search) methods. This approach requires an initial feasible point to work, which can be, as mentioned above, a Nash equilibrium. Computation of such a feasible point may be quite difficult, and so one might consider alternative approaches.

Using the penalty method [96] to penalize solutions in which costs do not reflect the current strategy, and introducing an additional decision variable z such that solutions are within z of equilibrium, (3.30) may be rewritten as,

$$\begin{aligned}
&\text{minimize } g^T x + \sum_{i=1}^{n_1} \rho_1 (x^T B_i y - x^T b_i)^2 + \rho_2 z \\
&\text{subject to} \\
&Ax = a, \\
&x^T C_j y - x^T c_j \geq -z, \quad j = 1, \dots, n_2,
\end{aligned}$$

$$\begin{aligned} z &\geq 0, \\ x &\geq 0. \end{aligned} \tag{3.32}$$

For sufficiently large ρ_1 and ρ_2 , (3.30) and (3.32) should yield equivalent solutions.

One potential solution method is to apply a nearly block coordinate descent algorithm [93] to (3.32). In this approach, (x, z) and (y, z) are alternately held fixed, and (3.32) is solved for the remaining decision variable. Additionally, the parameters ρ_1 and ρ_2 are slowly increased with each iteration. There are two main advantages to such an approach:

1. When $\rho_1 = \rho_2 = 0$, the solution to (3.32) is simply the globally optimal strategy, which can be obtained via linear programming.
2. Each iteration corresponds to an optimization problem over a convex feasible region, so standard convex optimization methods [96] apply.

The solution algorithm therefore begins at a global optimum, and terminates when a solution is reached that is satisfactorily close to equilibrium. One must be careful with such an approach however. If ρ_1 and ρ_2 are increased too quickly, the method may converge prematurely. It is also unclear whether a nondecreasing sequence of values for ρ_1 and ρ_2 can be found for every problem. That is, the effectiveness of such an approach rests on one's ability to increase ρ_1 and ρ_2 at each iteration.

3.5 Information Theoretic Metrics in NEOps

One of the basic principles of Network Enabled Operations is the ability to share information, which in the present model includes the capability of coordinating actions. In this section the specifics of coordinated execution of a correlated equilibrium, NEOps equilibrium strategy are investigated in detail. Section 3.5.1 investigates the amount of bandwidth required for such coordination, while Section 3.5.2 discusses associated network architectures.

3.5.1 Optimizing Information Flow in NEOps

In the last section, the NEOps equilibrium strategy was characterized as an optimal correlated equilibrium, given by the solution to a nonlinear program, (3.29). In this section, we quantify the amount of communication involved in the implementation of such a strategy by considering the mutual information between the strategies of the various players.

Measuring the amount of necessary communication is important for implementation, since the amount of communication bandwidth required versus the amount available determines feasibility of the NEOps solution.

We must make a distinction between types of communication here. Specifically, the focus here is on online communication, i.e. communication necessary for the implementation of a defensive strategy, as opposed to offline communication, which would be used to share utilities and jointly compute the strategy in the first place. Also disregarded is communication that would send information that could have been perfectly anticipated by the receiver. In other words, we seek the absolute minimum communication requirements for the implementation of a NEOps equilibrium strategy.

In the context of games, mutual information measures the reduction in uncertainty about some players' decisions given knowledge of others. Its definition, for a bivariate probability distribution, (see [99]) is given by:

$$I(X; Y) = \sum_{x,y} \mu(x, y) \log_2 \frac{\mu(x, y)}{\mu_X(x)\mu_Y(y)}, \quad (3.33)$$

where random variables X, Y have joint distribution μ and marginal distributions μ_X and μ_Y . Such random variables are taken here as representing the decisions of independent players.

Shannon's mutual information measure, above, is a "distance" measure; specifically it is the relative entropy between the joint strategy probability distribution and the product of the marginal distributions. The marginal distributions represent an approximation to the NEOps equilibrium strategy in which each player acts completely independently, without any coordination required. The mutual information "distance" is then representative of how much coordination is required to get from this independently executable (non-equilibrium) strategy to the NEOps equilibrium strategy itself.

To illustrate the connection between mutual information and communication necessary for implementing the NEOps equilibrium strategy, consider a three player (repeated) game, with Players One, Two and Three making decisions given by random variables X, Y , and Z with corresponding joint distribution f and marginals f_X, f_Y , and f_Z . If play is according to a Nash equilibrium (in either pure or mixed strategies), then f is a product distribution and clearly $I(X; Y, Z) = 0, I(Y; Z) = 0$, etc. That is, the mutual information between players is zero exactly when communication between players is not required, i.e. when they play a Nash equilibrium. Similarly, when the strategy being played is such that each player's action is determined completely by the other two actions, the mutual information

is maximized (e.g. anti-coordination games). The requirement for coordination increases with the mutual information in the system.

Although there are many quantities of interest, we focus on the mutual information between each player and their collective opponents. Let Z_i^l be the random variable representing the action of Player l in state i , and similarly let Z_i^{-l} represent the actions of all players but l . These quantities have marginal distributions

$$\mu_i^l(\cdot) = \sum_{u^{-l} \in U^{-l}(i)} \mu_i(\cdot, u^{-l}), \quad (3.34)$$

for Z^l , and,

$$\mu_i^{-l}(\cdot) = \sum_{u^l \in U^l(i)} \mu_i(u^l, \cdot), \quad (3.35)$$

for Z^{-l} . The quantities of interest may now be defined as

$$I(Z_i^l; Z_i^{-l}) = \sum_{u \in U(i)} \mu_i(u) \log_2 \frac{\mu_i(u)}{\mu_i^l(u^l) \mu_i^{-l}(u^{-l})}. \quad (3.36)$$

For convenience, we also define the maximal mutual information by

$$\mathcal{I}(Z_i) = \max_{l \in L} (I(Z_i^l; Z_i^{-l})). \quad (3.37)$$

The mutual information defined in (3.36) is closely linked to the amount of communication required by Player l to implement the NEOps equilibrium strategy μ in state i . Suppose, as an external observer, we are able to observe Z_i^l , the actions of Player l in each state i . We may think of this player as a noisy receiver, and the observations of his actions as received signals about the values of Z_i^{-l} , since the player chooses actions according to his information about Z_i^{-l} . The quantity $I(Z_i^l; Z_i^{-l})$ measures the reduction in uncertainty about Z_i^{-l} due to our knowledge of Z_i^l , and thus indirectly measures the amount of communication between Player l and the rest of the system.

There is a notion of *conservation of information* here; using Player l as a noisy receiver, an external observer can directly measure a reduction in uncertainty of the joint actions of other players and thus the amount of information communicated through Player l . Since there is no other source of information creation or loss, the inference is that the same amount of information is communicated to Player l .

The mutual information measure can be used as a design consideration. Under the as-

sumption that channel capacity is relatively uniform within a NEOps network, that is each ship l has channel capacity C , we have the design requirement $\mathcal{I}(Z_i) \leq C$ for all $i \in \mathbf{S}$. A correlated equilibrium is executable if and only if there exists sufficient communication capability, i.e. beyond C , between ships. If this requirement is not satisfied, then a constrained correlated equilibrium must be implemented. In the worst case, with $C = 0$, the task group implements a Nash equilibrium.

More generally, given bounds on the mutual information, a constrained correlated equilibrium can be computed by including an additional constraint in (3.29). For example, given bounds C^l on the mutual information of Player l , we may characterize the optimal correlated equilibrium as the solution to the following program.

$$\begin{aligned}
 & \text{minimize } \sum_{i \in \mathbf{S}'} \sum_{u \in U(i)} g(i, u) \pi(i, u) - \sum_{l \in L} \sum_{i \in \mathbf{S}'} \beta(i, l) J^l(i) \\
 & \text{subject to } (\forall i \in \mathbf{S}', l \in L, u^l, d^l \in U^l) : \\
 & \quad \sum_{u \in U(i)} \pi(i, u) = \sum_{j \in \mathbf{S}'} \sum_{u \in U(j)} p(j, i; u) \pi(j, u) + \frac{1}{|\mathbf{S}'|}, \\
 & \quad \sum_{z \in U(i)} \pi(i, z) J^l(i) = \sum_{u \in U(i)} g^l(i, u) \pi(i, u) + \sum_{j \in \mathbf{S}'} \sum_{u \in U(j)} p(i, j; u) \pi(i, u) J^l(j), \\
 & \quad \sum_{u^l \in U^l(i)} \sum_{u^{-l} \in U^{-l}(i)} \pi(i, (u^l, u^{-l})). \\
 & \quad \log_2 \frac{\pi(i, (u^l, u^{-l})) \sum_{z \in U(i)} \pi(i, z)}{\sum_{z \in U^{-l}(i)} \pi(i, (u^l, z)) \sum_{z \in U^l(i)} \pi(i, (z, u^{-l}))} \leq C^l \sum_{z \in U(i)} \pi(i, z), \\
 & \quad \sum_{u^{-l} \in U^{-l}(i)} \pi(i, (u^l, u^{-l})) \left[g^l(i, (u^l, u^{-l})) - g^l(i, (d^l, u^{-l})) \right] \leq \\
 & \quad \sum_{u^{-l} \in U^{-l}(i)} \pi(i, (u^l, u^{-l})) \sum_{j \in \mathbf{S}'} \left[p(i, j; (d^l, u^{-l})) - p(i, j; (u^l, u^{-l})) \right] J^l(j), \\
 & \quad \pi(i, u) \geq 0 \quad \forall u \in U(i).
 \end{aligned} \tag{3.38}$$

(3.38) is quite complex and we do not discuss it further here. Rather we remark that when each $C^l = 0$, the program represents a characterization of a stationary Nash equilibrium as the solution to an optimization problem, which is a rarity compared to the more common fixed point characterization. See [101] for examples of both. (3.38) is solved for a simple two player NEOps missile deflection game in the next section.

3.5.2 Implementation Architectures for NEOps Equilibrium Strategies

Up to this point, we have deliberately avoided any specific scheme for executing a stationary correlated equilibrium strategy. Indeed, the main objective of this chapter is to characterize these strategies without being bound to a specific implementation. This is due to our interpretation of a correlated equilibrium, not as a prescription for optimal play, but as a description of how players will eventually make decisions under NEOps, an interpretation inspired by [89], where randomization is considered to be an artifact of the uncertainty about actions held by an outside observer. In this section, this philosophy is temporarily abandoned, as we explore the question of how one might implement a given NEOps equilibrium strategy by explicitly simulating the randomness inherent in the strategy. We include this section because all currently known implementation methods for correlated equilibrium, as surveyed here, are concerned with implementing a prescribed correlated equilibrium, not with describing situations in which correlated equilibria spontaneously arise. For simplicity, we restrict our attention to static correlated equilibrium, with the understanding that a stationary (dynamic) correlated equilibrium can be implemented by considering an appropriate series of static (auxiliary) games.

Two approaches to implementing a correlated equilibrium can be found in the literature, those which require a central command and control station (Figure 3.2(a)-(c)), and those which do not (Figure 3.2(d)). Both will be discussed below.

The first approach includes the original, canonical approach as outlined as in [88] and depicted in Figure 3.2(a). Here, a central control station randomly draws joint actions from a known correlated equilibrium, and sends a private message to each player recommending only their component of the joint action. The players, aware of the distribution, find it rational to follow the recommendations. This method requires L separate channels, one for each player, and appears to be the most efficient approach, although it resembles a centralized implementation so much that it may be of limited use in practice.

An interesting alternative but equivalent approach is to implement an acceptance rejection method, in which players send proposed actions to the control station for authorization. Suppose that L players, or platforms, plus a central control station, wish to implement a known correlated equilibrium, defined by joint probability distributions μ , where the probability of jointly selecting action (y_1, y_2, \dots, y_L) is $\mu(y_1, y_2, \dots, y_L)$. Then the following acceptance rejection algorithm (see, for example, [116]), generates actions according to distribution μ .

Algorithm 3.5.1 • *Each player l randomly proposes an action Y_l according to the*

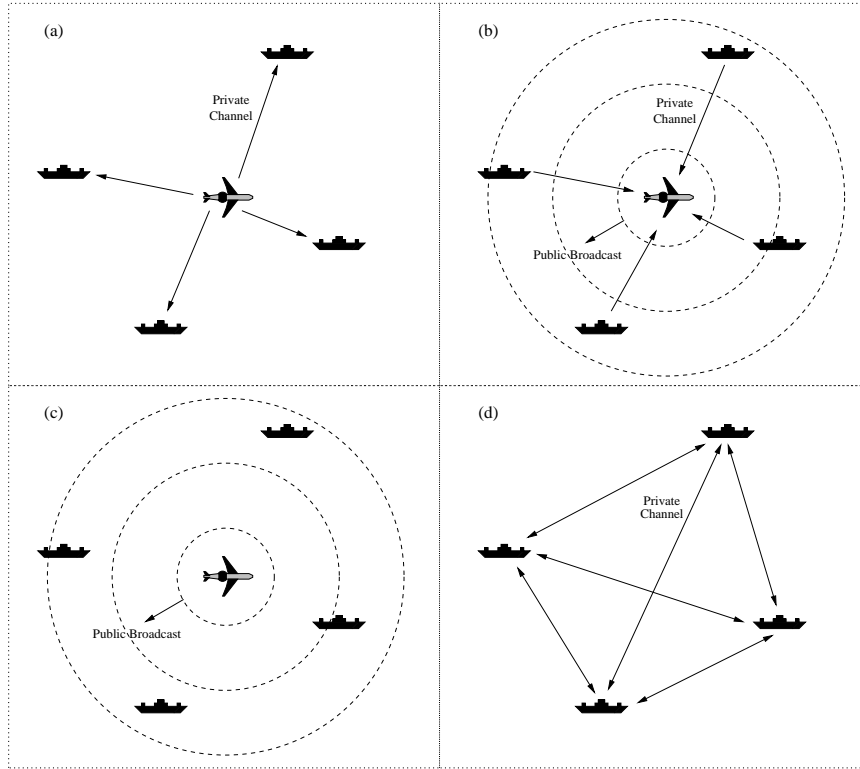


Figure 3.2: Communication architectures for implementing a NEOps equilibrium strategy. The aircraft represents a central command and control station. Private, public, or a combination of channels can be used for implementation, although private channels are required for optimal performance (a),(b),(d).

marginal distribution μ^l .

- Central control randomly accepts the joint action with probability

$$\frac{\mu(Y_1, Y_2, \dots, Y_L)}{c\mu^1(Y_1)\mu^2(Y_2)\dots\mu^L(Y_L)},$$

where

$$c = \max_{y_1, \dots, y_L} \left(\frac{\mu(y_1, y_2, \dots, y_L)}{\mu^1(y_1)\mu^2(y_2)\dots\mu^L(y_L)} \right).$$

- If the proposed joint action is accepted, the players implement it. Otherwise, they propose a new action.

It is well known that the average number of proposals needed in this scheme is c , which gives a rough measure of the amount of communication required to implement the joint strategy. The use of marginal distributions is justified since the product of marginal distributions is closest, in terms of Kullback-Leibler distance, to the true distribution μ while still being independently generatable by the players, (Lemma 13.8.1 of [99]). Since the mutual information is defined as the Kullback Leibler distance between μ and the product of its marginals, we have that the number of proposals increases with the mutual information between players, reinforcing the idea of Section 3.5.1 that mutual information is a good indicator of the communication overhead required to implement a correlated equilibrium. The acceptance rejection method requires two-way communication, and uses $L + 1$ channels, since the control station can simply broadcast its accept/reject message to all players (see Figure 3.2(b)). This is more bandwidth-intensive than Aumann's original scheme, but shares the computational burden throughout the system.

Interestingly, there is another central control station-dependent approach corresponding to the two-way private/public communication scheme of Figure 3.2(b). In [110], it was shown how such an approach can be used to implement any correlated equilibrium. First, each player sends a private message to the control station, then the control station broadcasts a public message that is a deterministic function of these private messages. Each player then selects an action as a function of the private message sent and the public message received. Again, $L + 1$ separate channels are needed for implementation. (However, only one round of communication is needed.)

In many cases, it is desirable to restrict one's attention to efficiently implementable correlated equilibria, for example when communication is highly restricted. The public broadcast implementations above and Aumann's original one-way private message implementation to-

gether suggest the efficiently implementable class of public correlated equilibria [104]. In a public correlated equilibrium, players observe a commonly broadcast, as opposed to a private, random variable, and choose a random action based on the outcome. However, the players themselves do not communicate with the control station (see Figure 3.2(c)). It is well known [104] that, using this approach, players can implement any correlated equilibrium whose payoffs are within, but not outside, the convex hull of Nash equilibrium payoffs. This is useful for implementing a joint strategy that is fair between all players, and is attractive when the optimal correlated equilibrium does not yield a significant global improvement over the set of Nash equilibria. The major benefit of using public correlated equilibria is that communication is only in one direction, and only a single broadcast channel is required, instead of $L + 1$ channels, as above.

We illustrate the implementation of a public correlated equilibrium for the case of two players, noting that the concept carries over to more players at the expense of more complex notation. Consider a two player game in which Player One's action is given by random variable X , and Player Two's action is given by Y . Let P be the probability distribution matrix of joint actions (X, Y) , and assume P is a correlated equilibrium distribution. Suppose P is decomposable as:

$$P = P_x^T D P_y, \quad (3.39)$$

where $P_x \mathbf{1} = \mathbf{1}$, $P_y \mathbf{1} = \mathbf{1}$, (i.e. $\mathbf{1}$, the column vector with components all equal to one, is an eigenvector with eigenvalue 1) and D is diagonal probability distribution matrix. It is straightforward to show that

$$P = P_x^T D P_y = \sum_{b=1}^{N_b} D(b, b) P_x^T(b, \cdot) P_y(b, \cdot), \quad (3.40)$$

where N_b is the number of signals. That is, P in this case is decomposable into a convex combination of product distributions. In this case the correlated equilibrium can be implemented as follows. First, a control station broadcasts a random variable B such that

$$\mathbf{P}(B = b) = D(b, b). \quad (3.41)$$

Players One and Two then choose actions randomly according to the probability distribution given by row B of P_x and P_y , respectively. (We say that actions X and Y are conditionally independent, given B .)

The set of such decomposable correlated equilibria is nonempty; any convex combina-

tion of Nash equilibria qualifies. That is, if the rows of P_x and P_y are Nash equilibrium distributions, then P is a correlated equilibrium for any diagonal matrix D . It is interesting to note that certain convex combinations of non-Nash equilibria may also yield a correlated equilibrium, but in this case it is not clear that one can induce the players to play such an equilibrium in the same manner, since there will be a temptation for each player to deviate once the public signal is observed.

Restricting our attention to convex combinations of Nash equilibria, we can compute the optimal public correlated equilibrium for a given game. For example, the optimal Egalitarian, cost minimizing public correlated equilibrium can be computed as follows:

1. compute all n Nash equilibria and set the resulting strategies as the rows of P_x and P_y .
2. Define cost vectors C^1 and C^2 corresponding to each Nash equilibrium. If U^1 and U^2 are the cost matrices of each player, C^i can be obtained as the diagonal of $P_x U^i P_y^T$. Define $C = \max(C^1, C^2)$.
3. Solve the optimization problem

$$\begin{aligned}
 & \text{minimize } C^T x \\
 & \text{subject to:} \\
 & \mathbf{1}^T x = 1 \\
 & x \geq 0
 \end{aligned} \tag{3.42}$$

4. Set $D(i, i) = x(i)$ for $i = 1, 2, \dots, n$.

Here $\mathbf{1}$ is the column vector with all components equal to 1. This is a particularly simple convex optimization problem, and the resulting equilibrium is easily implementable, as noted above. For the NEOps missile deflection game, an aircraft or command ship (e.g. AWACS aircraft, aircraft carrier) can be used as a control station to broadcast random values to each ship, facilitating a fair public correlated equilibrium.

In fact, a control station is not strictly necessary for implementation of a public correlated equilibrium. If players are equipped with a list of common random numbers, then, using a synchronized clock, they may simulate any common random variable themselves. In this case, the correlation between players is implicitly provided by the common observation of the time at which the game is to be played. For this reason, public correlated equilibria are often identified with sunspot equilibria. (Since we are dealing with a stochastic game,

we could in fact be even more clever than this, by harnessing the randomness inherent in state transitions as public correlation signals! However, this would only yield suboptimal or sub-fair equilibria in all but a very special class of games.)

This observation of control station-independence leads us to consider the second approach to implementing a correlated equilibrium. This approach allows players to implement an equilibrium without the need for a control station, by communicating only amongst themselves (Figure 3.2(d)). The literature here is extensive, as many different algorithms have been proposed for implementing a correlated equilibrium without a control station, for example in [90, 92, 102, 105, 120, 121].

Such methods generally require extensive message passing, referred to as *cheap talk*, to ensure that players are honestly and reliably playing the correlated equilibrium. Players must transmit random variables to each other in such a way that no player has full information about the joint action to be played, and such that deviations from the protocol can be detected and punished. Understandably, this message passing phase is often extensive, and we conclude that these methods too inefficient for our purposes at present.

Regardless of the particular implementation, each correlated equilibrium requires a certain minimal amount of communication for implementation. The purpose of this chapter is to examine the trade-off between communication and performance, not to identify optimal communication schemes for the execution of a correlated equilibrium. Nonetheless, a survey of implementation methods allows us to grasp the richness of the class of candidate communication schemes.

3.6 Numerical Example of NEOps

In this section we derive and analyze NEOps equilibrium strategies for a multiple-ship, multiple-missile CM scenario, and highlight some of the fundamental aspects of the decentralized missile deflection problem. A NEOps equilibrium strategy is computed using standard nonlinear programming techniques, and the resulting communication requirements, as measured by the maximal mutual information quantity $\mathcal{I}(Z_i)$ of (3.37), are computed and analyzed.

The missile deflection scenario is as follows. M missiles initially threaten three ships. Each missile has two modes of operation: a target acquisition mode and an attack mode. In the target acquisition mode, the missile attempts to select one of the three ships for attack. In the attack mode, it homes in on this ship using radar. The ships are equipped with various electronic countermeasures, which can be deployed against only one missile at

a time. In the target acquisition mode, these countermeasures are simple; they reduce the ship's radar visibility to the selected missile. In the attack mode, the countermeasures are designed to confuse the missile's targeting system, by providing false target information.

To decrease complexity, we model the scenario by an inner and outer game. In the inner game, we consider a single missile in attack mode, defended against by a subset of the three players. In the outer game, we consider the problem of selecting which missiles to engage. The equilibrium strategy outcomes of the inner game are used to define the costs in the outer game. This is a similar approach to that of a multiple time-scale Markov decision processes [98].

After describing the inner game, the NEOps equilibrium strategy for $M = 3$ missiles is derived and analyzed. To illustrate how the model can be adapted to high-dimensional problems with low computational complexity, we also compute a myopic NEOps equilibrium strategy for $M = 10$ missiles. The NEOps equilibrium strategy is compared to the optimal and Nash equilibrium strategies.

3.6.1 Inner Game (Single Missile in Attack Mode)

In this section, the modeling notation of Section 3.2. In the inner game, a single missile moves from left to right across an 11×11 battlespace B , towards one of eleven target positions at $T = \{(x, y) \in B : x = 11\}$. The initial target of the missile is at coordinates $\mathfrak{T}^1 = \{(11, 6)\}$ and belongs to Player One. The other two ships are at coordinates $\mathfrak{T}^2 = \{(11, 1)\}$ and $\mathfrak{T}^3 = \{(11, 11)\}$ and belong to Players Two and Three, respectively. The true target set is $\mathfrak{T} = \mathfrak{T}^1 \cup \mathfrak{T}^2 \cup \mathfrak{T}^3$. A specified subset of ships deploy countermeasures to defeat the missile's targeting system.

The missile's position and target evolve according to a Markov chain $\{X_t^m, Z_t^m, t = 1, 2, \dots\}$ as in Section 3.2.1, where X_t^m evolves on B and Z_t^m evolves on T . At each time index, the missile moves one unit in the positive x direction, and moves one unit randomly in the y direction with probability proportional to the angle induced between its current (x, y) -position and the (x, y) -position of its current target location. That is, X_{t+1}^m depends on the angle between X_t^m and Z_t^m , as shown in Figure 3.3 via (3.4). A small amount of noise is added to simulate random deviations in the missile guidance system.

The missile's current target location evolves according to a complicated transition probability matrix, controlled by the current joint CM deployed. Player One may deploy CM L or R , intended to cause the target position to drift to the left or right. Players Two and Three may each deploy CM A or I , to increase own attractiveness as a target, or to increase the general amount of radar interference, respectively. All players also have a null action

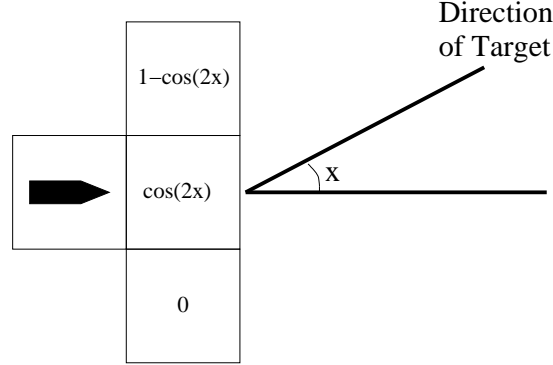


Figure 3.3: Physical missile transition probabilities as a function of target angle for the NEOps example scenario.

N available.

For each of the 27 resulting joint CM actions, the missile dynamics are outlined below. Note that the dynamics are different depending on whether $Z_t^m \leq (11, 2)$, $(11, 3) \leq Z_t^m \leq (11, 9)$, or $Z_t^m \geq (11, 10)$. For each case, if the joint CM fails, Z_{t+1}^m will revert to \mathfrak{T}^2 , \mathfrak{T}^1 , or \mathfrak{T}^3 , respectively. That is, the missile tends to stick with its nearest target. When $Z_t^m \leq (11, 2)$, three types of joint CM action have a high likelihood of success: (1.) Player Two plays A , Player One plays L , and/or Player Three does not play A . (2.) Player One plays L or R , and all others play N . (3.) One of Player Two or Three plays I and Player One does not shift the target in that player's direction. The situation is similar for $Z_t^m \geq (11, 10)$. When $(11, 3) \leq Z_t^m \leq (11, 9)$, joint CM is effective in two situations: (1.) One of Player Two or Three plays A , and Player One attempts to shift the target in that direction. (2.) One of Player Two or Three plays A while the other plays N , and Player One does not shift the target in that player's direction. When CM is effective, the target Z_{t+1}^m tends to shift in the desired direction by a random amount (the actual distribution is a modeling detail). Otherwise Z_{t+1}^m moves to the nearest available true target.

Each possible subset of participants defines a particular inner game, for which we compute a NEOps equilibrium strategy. Objectives are as in (3.14) and (3.21). The complex interdependencies in the controlled missile dynamics are chosen so that each inner game has at least one effective strategy which must be achieved through cooperation. On the other hand, since each participating player hopes to direct the missile to the spaces between ships while avoiding danger, cooperating while satisfying the equilibrium conditions is not trivial.

The game is actually of the finite horizon type; it ends when the missile reaches the right

Participation	P1 Hit Prob.	P2 Hit Prob.	P3 Hit Prob.
P1,P2,P3	.0108	.0166	.0182
P1,P2	.0141	.0198	.0353
P1,P3	.0141	.0353	.0198
P2,P3	.0563	.0183	.0179
P1	.1553	.0292	.0292
P2	.0996	.0251	.0437
P3	.0996	.0437	.0251
None	.5543	.1383	.1383

Table 3.1: Probability of hit for each player P1,P2,P3 in each inner game (attack mode) under the NEOps equilibrium strategy.

edge of the state space or a target. This makes computation of the correlated equilibrium particularly simple because (3.29) can be solved by backwards induction. That is, instead of solving 1331 games simultaneously, they may be solved one at a time to compute the NEOps equilibrium; the costs in any game depend only on the policy in future games.

The NEOps equilibrium strategy is reasonable, although too complicated to describe in detail here. In general, players jointly select CM actions so as to drive the missile away from the flotilla, with each player particularly concerned about their own welfare. For example, when all players participate and $(X_t^m, Z_t^m) \in \{((x, 6), \mathfrak{T}^1), 1 \leq x \leq 11\}$ (the missile is heading directly for Player One), the players choose $u_t = (R, I, A)$ with probability one to shift the target to the area between Players One and Three. This occurs for $x \neq 8$. For $x = 8$, the ships enforce a compromise in which $E[Z_{t+1}^m] = 6$, but with large variation. This is the best compromise, since at $x = 8$ all ships are highly vulnerable and must maximize their own safety. Note also that the communication peaks at this point (Figure 3.4). The NEOps strategy in other states follows similar logic; players succeed in coordinating, but at no time is there self sacrifice.

Achieving the NEOps equilibrium strategy requires communication, since players must often avoid joint actions that would cancel each other out, leaving at least one player vulnerable. The communication overhead, as reflected by the mutual information between player actions, is indicated in Figure 3.4.

The NEOps equilibrium strategy was computed for eight inner games, corresponding to each possible subset of players. The probabilities of hit are shown in Table 3.1. These values assume a uniform distribution over initial states (i.e. $X_1^m = (1, y)$), and are used as parameters in the outer game below.

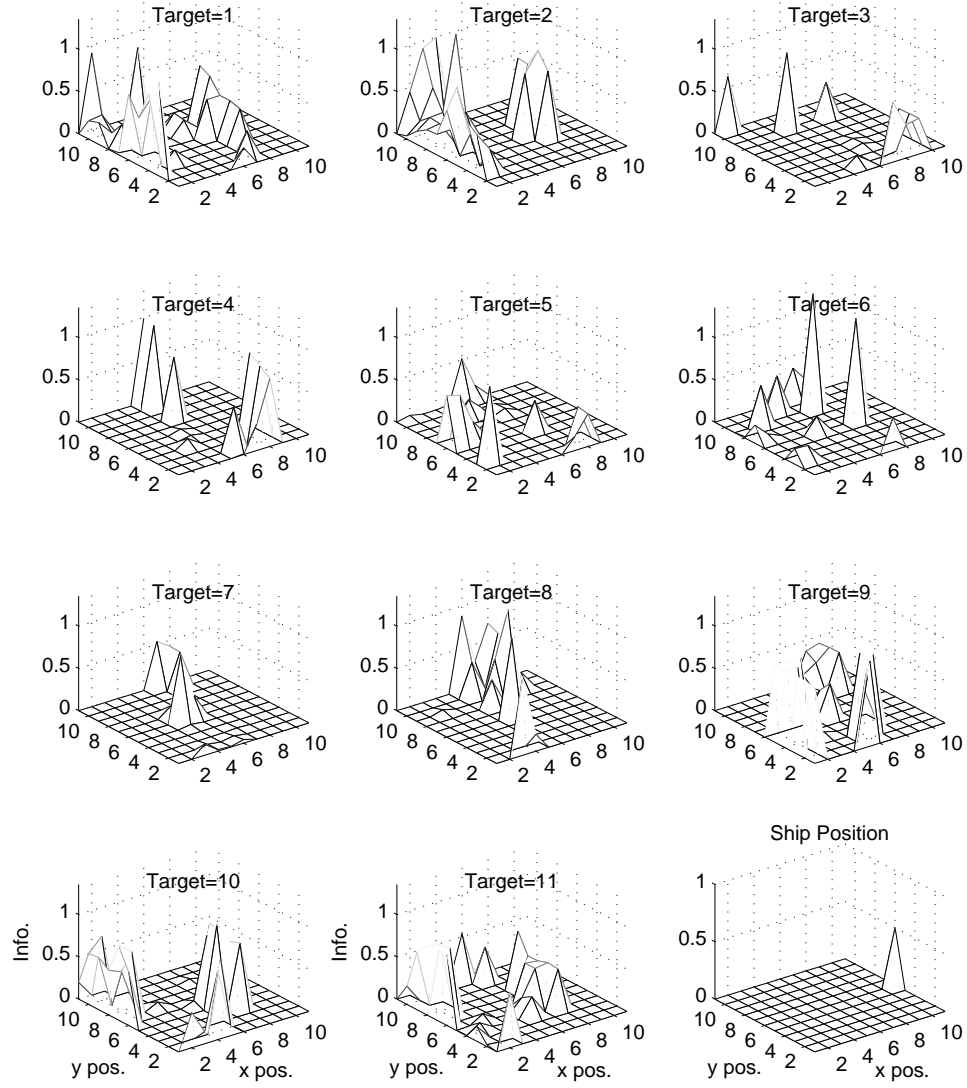


Figure 3.4: Maximum mutual information for inner game with all players participating.

3.6.2 Outer Game (Selection from Three Missiles)

In the first outer game, $M = 3$ missiles threaten the three ships. Each missile state is given by its mode (target acquisition, attack mode against Player l , terminal state). The three ships play a stochastic game over 125 states, corresponding to all possible combinations of states of the three missiles. In each state, each of the three ships must choose missile to engage. The associated state-action costs and transition probabilities are determined by considering the effect of these decisions on each missile separately. The total state-action costs are then obtained by adding the cost contribution of each missile, while the transition probabilities are obtained by treating the missiles independently on the product space as in Section 3.2.2.

The single-missile costs and transition probabilities differ depending on the missile mode and the set of players deploying CM against it. The values are outlined below for each case.

Missile in Target Acquisition Mode The transition probabilities of a missile in target acquisition mode are given according to the following rules: (1.) If no ship selects the missile, it enters attack mode against any player with equal probability 0.2, and otherwise returns to acquisition mode with probability 0.4. (2.) If one ship selects the missile, that ship is targeted with probability 0.1, while the other two are targeted with probability 0.25 each. The missile terminates or returns to acquisition mode with equal probability 0.2. (3.) If two ships select the missile, the termination probability increases to 0.3. Each of the participating ships is targeted with probability 0.25, while the non-participating ship is targeted with reduced probability 0.1. (4.) If all three ships select the missile, each ship is targeted with probability 0.25, and the missile terminates with probability 0.25. In each case, the immediate cost contribution is zero, since the missile cannot directly reach a ship in acquisition mode.

Missile in Attack Mode When a missile enters attack mode, and a subset of players deploy CM against it, one of the inner games is played. By convention, the missile always attacks Player One in the inner game, so a relabeling of players may be needed. The ship being targeted becomes Player One, while among the other two ships, the lower numbered ship becomes Player Two, and the remaining ship becomes Player Three for the duration of the inner game. The outcome is then determined by the appropriate row in Table 3.1. Specifically, the cost contribution to each player is the probability of a missile hit to that player, and the probability of transitioning to the terminal state is the sum of these hit probabilities. That is, the players act to minimize the expected probability of hit from that

missile. The remaining transition probability q is divided as follows. (1.) If a ship engages a missile alone, it will not be targeted again in the next stage. Instead, each of its sister ships will be targeted with probability $q/6$, and the target acquisition mode will be reached with probability $2q/3$. (2.) If two ships engage a missile, they are each targeted again in the next stage with probability $q/12$, while the remaining ship is targeted with probability $q/6$, and the target acquisition mode is entered with probability $2q/3$. (3.) If all three ships engage a missile they are each targeted again in the next stage with probability $q/16$. The remaining mass $13q/16$ is the probability of re-entering the target acquisition mode.

Comparison of Strategies We compute and compare the Nash equilibrium, NEOps equilibrium and optimal strategies for the outer game. The total expected number of missile hits for these strategies were found to be 1.10, 0.96, and 0.72, respectively. In other words, the NEOps equilibrium strategy offers a 37.5% relative improvement over the Nash equilibrium.

Comparing the strategies in detail reveals some fundamental differences. Most strikingly, when a missile m threatens player l , it turns out to be optimal for l to ignore m , while a player $k \neq l$ targets m for CM. However, in the Nash and NEOps strategies, l will always deploy CM against m . On the other hand, the NEOps strategy is more cooperative than the Nash, since ships assist each other much more often in the former. In addition, the NEOps strategy eliminates obvious miscoordination present in the Nash case, in which ships may occasionally select a missile that is already terminated.

Comparing the strategies line for line, we discovered the NEOps equilibrium strategy was much closer to the optimal strategy than the Nash. The degree of similarity, measured as the common area under two probability distributions, is 0.26 for the NEOps and optimal strategy, and only 0.09 for the Nash and optimal strategy. In fact, since the optimal strategy computed is a maximum entropy version (it randomizes between all alternatives when it is indifferent), versions of the optimal strategy surely exist with much higher degrees of similarity (to both the Nash and NEOps strategies). We also found that the NEOps strategy was closer in terms of Kolmogorov-Smirnov distance, and in terms of Kullback-Leibler distance on support restricted to the optimal strategy.

The average communication overhead for the NEOps equilibrium strategy is 0.3644 bits per state. Since outer game actions are made much less frequently than inner game ones, this overhead can be accommodated using a very small, but nonzero bandwidth.

Participation	Cost to P1	Cost to P2	Cost to P3
P1,P2,P3	.012x	.012x	.012x
P1,P2	.02x	.02x	.005x
P1,P3	.02x	.005x	.02x
P2,P3	.005x	.02x	.02x
P1	0	.011x	.011x
P2	.011x	0	.011x
P3	.011x	.011x	0
None	.01x	.01x	.01x

Table 3.2: Amended cost contribution for a missile in target acquisition mode.

3.6.3 Outer Game (Myopic Selection from Ten Missiles)

In this section we illustrate the applicability of our results to a much larger game model. The situation is the same as in Section 3.6.2, but now we consider an attack by $M = 10$ missile simultaneously. Since it is unrealistic to perform dynamic programming on such a large state space (5^{10} states), we restrict our attention to myopic policies. That is, we assume that the expected future cost in each state is zero, so that ships choose actions based only on their current state-action costs. We then compare the resultant myopic NEOps equilibrium strategy with the myopic optimal strategy for several states of interest.

It should be noted that the myopic strategies are generally worse than the true optimal strategies, but may perform reasonably well. In the three missile example above, the myopic NEOps equilibrium strategy yields a total expected number of hits of 0.98 instead of 0.96, whereas the myopic optimal strategy gives 1.03 instead of 0.72.⁴ Other benefits of considering myopic policies are that no model of the missile dynamics (between inner games) is required, and the computational complexity is drastically reduced.

A feature of the myopic policies is that, since no cost is associated with missiles in target acquisition mode, ships will tend to disregard those threats. We amend this by adjusting the cost contributions in acquisition mode to those of Table 3.2. The model parameter x is initially set to one. The total cost contribution is minimized when exactly one player targets such a missile. However, the effect is small enough (for $x \approx 1$) that if other threats are present, they take priority.

We proceed to analyze the myopic strategies for a few key states, and then investigate how these strategies vary as x is increased. We label states as 10-tuples, with entry m

⁴It is unusual that the optimal strategy performs worse than equilibrium, but possible considering the strategies are measured by a different criteria than that which they are optimizing with respect to.

corresponding to the state of the m^{th} missile. State value 0 corresponds to target acquisition mode, values $1 \leq i \leq 3$ correspond to the missile targeting Player i , and value 4 corresponds to the terminal state.

First, for states with all entries equal to either 0 or 4, the myopic NEOps and optimal strategies are identical; each missile in acquisition mode is targeted by exactly one ship for CM. This requires some coordination, which decreases as the number of missiles increases.

For other states, the myopic NEOps equilibrium strategy behaves according to the following basic threshold rule: When a ship is threatened by $k \leq 1$ missiles, its first priority is to target any missile that is threatening a *different* ship; when a ship is threatened by $k \geq 2$ missiles, its first priority is to target one of the missiles threatening *itself*. For $x \approx 1$, missiles in state 0 are only targeted after these priorities are satisfied. This type of behaviour makes sense from inspection of Table 3.1. This is in contrast to the myopic optimal policy, in which a ship always prefers to target the missiles threatening different ships, regardless of the number of threats against itself. These policies require some degree of information exchange between the players, usually less than 1 bit per play.

For example, in State $(1, 2, 3, 0, \dots, 0)$ the equilibrium and optimal action is for each player to randomly select a missile other than the one threatening it for CM. However, in State $(1, 2, 3, 3, 0, \dots, 0)$ the equilibrium deviates from the optimal action in that Player 3 always selects Missile 3 or 4 for CM. This is in keeping with the equilibrium conditions.

When x is increased beyond a certain threshold, missiles in State 0 become more threatening and are hence taken more seriously. In general this threshold depends on the missile state; if it is exceeded, the ships tend to reorganize their behaviour so that, with positive probability, exactly one ship deals with each threat in State 0. So, for example, in State $(1, 2, 0, \dots, 0)$, if $x < 3.5$, Ship 3 randomly selects either Missile 1 or 2, for CM whereas as x increases beyond 3.5, Ship 3 begins to select from missiles 3 to 10 with increasing probability. In contrast, the optimal policy also displays this effect, but at a larger threshold. Therefore, for each state, there is a range of x in which the equilibrium policy is suboptimal.

Overall, even the myopic strategies appear reasonable, and even display a coherent structure, even if they cannot be expected to perform as well as the optimal strategies. By varying the parameters, we have shown that the myopic NEOps equilibrium strategy does not always coincide with the optimal strategy. Finally, analysis of the strategies shows that the amount of communication overhead is nonzero, but does not appear to require significant bandwidth in practice. The information exchange as measured by (3.37) varies by state, but is always smaller than 1 bit per state.

3.7 Conclusion

We have shown how the problem of decentralized deployment of electronic countermeasures against radar guided anti-ship missiles can be formulated as a transient stochastic game. When the decision makers in this game are linked through a communication network, the joint deflection strategy is given by an optimal stationary correlated equilibrium, referred to as a NEOps equilibrium strategy. Under this strategy, ships are able to simultaneously minimize their own probability of a missile hit as well as the probability of a missile hit to the entire task group.

The NEOps equilibrium strategy, which represents the collectively best strategy acceptable to all players, can be computed via the nonlinear optimization problem (3.29). Independent, rational decision makers with well defined local and global objectives, such as those in the missile deflection game presented here, would find it optimal to follow this strategy given that they are able to coordinate their actions through communication channels. In general, the strategy appears to be randomized, which is a natural property of the solution to an optimization problem with such a large number of constraints, but this randomization can also be viewed as a consequence of collective uncertainty associated with any equilibrium solution. Players are uncertain about how others will act, how others think they will act, and so on. This appears to be a fundamental consequence of the fact that players must simultaneously execute a single joint action while working to achieve differing objectives, and cannot be overcome unless either the simultaneity of action or multiple objective conditions are dropped. Communication can improve, but not dispel this state of affairs.

In a more general setting, self-interested systems capable of synchronizing observations and coordinating efforts are quite common. Manufacturing supply chains, biological systems, and data networks are just a few examples of such systems. It is hoped that the treatment presented here will serve as an example for similar study in such diverse areas.

We have also been able to describe a key performance measure, namely the communication overhead required for implementing a NEOps equilibrium strategy, via elementary concepts in information theory. Specifically, we have shown that the mutual information between a player's probability distribution over actions and the probability distribution over his opponent's actions is intimately tied to the amount of communication required by that player in the implementation of the NEOps equilibrium strategy. This relationship provides a measurable quantity for determining whether a given strategy can be feasibly implemented by a group of players with certain bandwidth limitations, or whether a lesser correlated or

even Nash equilibrium must be implemented instead. The formulation of communication requirements for implementing a NEOps equilibrium strategy leads to a trade-off between communication and performance at equilibrium.

The intuitive notion that cooperation and communication can improve performance among groups of linked, dynamic decision makers is central and important in modern systems. We have given a concrete example that illustrates that the link between communication and performance exists, but is nontrivial, and have demonstrated how the analytical tools of information theory and game theory can be used to analyze optimal, cooperative strategies.

Bibliography

- [86] D. Alberts, J. Garstka, and F. Stein. *Network centric warfare: developing and leveraging information superiority*. CCRP Publications, second edition, 1999.
- [87] E. Altman. *Constrained Markov decision processes*. Chapman and Hall/CRC, 1999.
- [88] R. Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1:67–96, 1974.
- [89] R. Aumann. Correlated equilibrium as an expression of bayesian rationality. *Econometrica*, 55(1):1–18, 1987.
- [90] R.J. Aumann and S. Hart. Long cheap talk. *The Hebrew University of Jerusalem, Center for Rationality*, 2002.
- [91] S. Babcock. Canadian network enabled operations initiatives. *Proceedings of the 9th International Command and Control Research and Technology Symposium*, 2004.
- [92] Elchanan Ben-Porath. Correlation without mediation: Expanding the set of equilibrium outcomes by "cheap" pre-play procedures. *Journal of Economic Theory*, 80(1):108–122, 1998.
- [93] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, second edition, 1999.
- [94] D. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1 and 2. Athena Scientific, second edition, 2001.
- [95] S. Blackman and R. Popoli. *Design and analysis of modern tracking systems*. Artech House, 1999.
- [96] S. Boyd. *Convex Optimization*. Cambridge University Press, 2004.
- [97] A. Cahn. General procedures leading to correlated equilibria. *International Journal of Game Theory*, 33(1):21–40, 2004.

- [98] H. Chang, P. Fard, S. Marcus, and M. Shayman. Multitime scale markov decision processes, 2002.
- [99] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley, 1991.
- [100] H. Eustace, editor. *The International Countermeasures Handbook*. EW Communications, sixth edition, 1980.
- [101] J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer-Verlag, 1997.
- [102] Francoise Forges. Universal mechanisms. *Econometrica*, 58(6):1341–64, 1990.
- [103] Francoise M Forges. An approach to communication equilibria. *Econometrica*, 54(6):1375–85, 1986.
- [104] D. Fudenberg and J. Tirole. *Game theory*. MIT Press, 1991.
- [105] Olivier Gossner. Secure protocols or how communication generates correlation. *Journal of Economic Theory*, 83(1):69–89, 1998.
- [106] A. Greenwald and K. Hall. Correlated-q learning. *Proceedings of the Twentieth International Conference on Machine Learning*, 2003.
- [107] S. Hart and A. Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.
- [108] S. Hart and A. Mas-Colell. *A reinforcement procedure leading to correlated equilibrium, in Economic Essays*, pages 181–200. Springer, 2001.
- [109] P. Herings and R. Peeters. Stationary equilibria in stochastic games: structure, selection and computation. *METEOR Working Paper No. RM/00/31.*, 2000.
- [110] Ehud Lehrer and Sylvain Sorin. One-shot public mediated talk. *Games and Economic Behavior*, 20(2):131–148, 1997.
- [111] M. Maskery and V. Krishnamurthy. Game-theoretic missile deflection in network centric warfare. *Proceedings of the IEEE International Conference on Networking, Sensing and Control*, 2005.
- [112] R. Nau, S. Gomez Canovas, and P. Hansen. On the geometry of nash equilibria and correlated equilibria. *International Journal of Game Theory*, 32(4):443–453, 2004.

- [113] A. Nowak and T. Raghavan. Existence of stationary correlated equilibria with symmetric information for discounted stochastic games. *Math. Oper. Res.*, 17(3):519–526, 1992.
- [114] Michal Pioro and Deepankar Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann Publishers, 2004.
- [115] M. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, 1994.
- [116] S. Ross. *Introduction to probability models*. Academic Press, sixth edition, 1997.
- [117] L. Shapley. Stochastic games. *Proceedings of the National Academy of Sciences USA*, 39(10), 1953.
- [118] H. Sherali, Y. Lee, and D. Boyer. Scheduling target illuminators in naval battle-group anti-air warfare. *Naval Research Logistics*, 42, 1995.
- [119] E. Solan and N. Vieille. Correlated equilibrium in stochastic games. *Games and Economic Behavior*, 38(2):362–399, 2002.
- [120] V. Teague. Selecting correlated random actions. *Proceedings of Financial Cryptography*, 2004.
- [121] T. Rabin Y. Dodis, S. Halevi. A cryptographic solution to a game theoretic problem. *Proceedings of CRYPTO*, 2000.

Chapter 4

Decentralized Activation in a ZigBee-enabled Unattended Ground Sensor Network ⁵

4.1 Introduction

Tasks such as detailed environmental monitoring and intruder detection depend on the ability to reliably observe highly localized events within a large coverage area. In the absence of highly sensitive equipment, researchers have proposed deployment of *unattended ground sensor networks* (UGSN) to provide this capability, by densely deploying short-range detection equipment throughout the region in question [122, 129, 137, 144, 145, 149, 150]. The challenge then is to coordinate the activity of a large number of devices “in the field” as efficiently as possible, so as to satisfy tight power and bandwidth constraints.

Since sensor coordination is a costly and complex process, we take a decentralized approach by modeling each sensor as an independent, nonlinear adaptive filter, which optimizes sensor performance through a utility function, which trades off the energy cost of acquiring data against its value. The adaptive filters control local functionality only, with minimal message passing, thereby reducing intra-sensor communication and associated energy costs. This decentralized approach is attractive because it facilitates self-configuration of sensors, which allows for fast deployment, as well as network robustness and scalability. Self configuration is achieved by the standard approach of improving filter performance based on time-averaged feedback.

The main idea of this paper is the following: Suppose each adaptive filter in the UGSN greedily optimizes its utility according to a simple adaptive rule (described below). In this case, *What can one say about the global network performance of the UGSN?* Our main

⁵A version of this chapter has been submitted for publication. Maskery, M. and Krishnamurthy, V. Decentralized Activation in a ZigBee-Enabled Unattended Ground Sensor Network, IEEE Transactions on Signal Processing.

results, Theorems 4.3.1 and 4.3.2, show that the sensor network's performance converges to the set of *correlated equilibria* of the underlying noncooperative game, which can be tracked as the system changes due to moving targets ⁶.

The set of correlated equilibria (see [125]) is arguably the most natural attractive set for decentralized adaptive algorithms such as the one considered here, and describes a condition of competitive optimality between sensors. Indeed, Hart and Mas-Colell observe, while describing such adaptive algorithms in [134] that:

“At this point a question may arise: Can one actually guarantee that the smaller set of Nash equilibria is always reached? The answer is definitely “no.” On the one hand, in our procedure, as in most others, there is a natural coordination device: the common history, observed by all players. It is thus not reasonable to expect that, at the end, independence among players will not obtain [be realized]. On the other hand, the set of Nash equilibria is a mathematically complex set (a set of fixed-points; by comparison, the set of correlated equilibria is a convex polytope), and simple adaptive procedures cannot be expected to guarantee the global convergence to such a set.”

Moreover, the coordination allowed by a correlated equilibrium can lead to higher performance than if each sensor was required to act in isolation.

Our main results in this paper are as follows:

- A game theoretic model for sensor activation is presented, based on energy and performance characteristics of the ZigBee communications protocol. The model allows individual sensors to trade-off their contribution to end-user awareness with the energy cost of activation.
- A computationally efficient (see Section 4.2.3) decentralized, adaptive filtering rule is given for activation control, which converges to the correlated equilibrium set of the underlying game. A constant stepsize allows for tracking of the equilibrium as the environment changes (e.g. due to moving targets or sensor outage). In stochastic optimization problems, the usual tool for proving convergence of a stochastic approximation algorithms to an optimum, is the ordinary differential equation approach [140].

⁶Robert J. Aumann was awarded the 2005 Nobel prize in economics. The following extract is from the Nobel Prize press release in October 2005: “Aumann also introduced a new equilibrium concept, correlated equilibrium, which is more general than Nash equilibrium, the solution concept developed by John Nash, an economics laureate in 1994. Correlated equilibrium can explain why it may be advantageous for negotiating parties to allow an impartial mediator to speak to the parties either jointly or separately, and in some instances give them different information.”

However, in the game theoretic setting of this paper, the network performance converges to the *set* of correlated equilibria. It is necessary to use differential inclusions (rather than ordinary differential equations) to analyze the convergence and tracking properties of the algorithm.

- An important aspect of the decentralized stochastic approximation algorithms used in this paper is that they require an estimate of how many neighboring sensors are active. We present an algorithm that exploits the CSMA structure of ZigBee/IEEE 802.15.4 for computing a maximum likelihood estimate of the number of nearby active sensors, to be taken as input by each adaptive filter. This is critical for estimating sensor interaction for the decentralized adaptive filtering rule.
- A timer-based pricing mechanism for load-sharing between sensors, which prolongs effective network lifetime by reducing energy consumption. Fairness is enforced by forcing active sensors into a standby mode after a sufficient period.
- We implement the decentralized algorithms on a ZigBee test-bed in Matlab and illustrate their performance. The ZigBee compatible sensor network architecture is described in Appendix A.

We favor a game theoretic approach due to the interdependence of individual nodes. The channel quality, required packet transmission energy and usefulness of a sensor's information all depend on the activity of other sensors in the network. Thus each adaptive filter must observe and react to other sensors' behaviour in an optimal fashion. This sets up a dynamic system in which adaptive filters and their environment (other filters/sensors) adapt to *each other*, instead of regarding the environment as a non-adaptive entity.

The simple decentralized adaptive filtering rule is based on the recent economics literature on regret-based algorithms, initiated in [133, 134]. At successive times, each filter/sensor chooses its actions based on its performance history. By suitably randomizing amongst the set of "best" responses to this history, play converges to the set of correlated equilibria, guaranteeing local optimality for each sensor node. A useful approach is to formulate this rule as a decentralized stochastic approximation deployed by the UGSN, which uncovers relations to similar problems in reinforcement learning for stochastic control [128], as well as optimization and parametric identification problems (e.g., recursive maximum likelihood and recursive expectation maximization algorithms; see [139]); see [127, 140] for excellent expositions of this area.

Game theoretic analyses of decentralized problems in UGSNs are becoming increasingly

common, see [131, 142] for a general overview. Other instances include, multihop routing [138], random access over a common channel [143], transmission power control [151], and topology control [141]. The common approach in these papers is to define a utility function that each system component selfishly maximizes, and then analyze system performance at the Nash equilibrium point of the resulting game. What has not been studied is games in which sensor nodes must decide how to sense their environment, as opposed to how to transmit their information. In addition, research so far has used only the Nash equilibrium concept; no attention is paid to correlated equilibria. We attempt to fill these gaps in our game theoretic sensor node activation approach.

The paper is organized as follows. Section 4.2 presents the game theoretic approach to the sensor activation control problem. Sensor awareness, utility, and an adaptive procedure for increasing individual sensor utility and the resulting overall utility are discussed. Section 4.3 provides a detailed analysis and convergence of the adaptive filtering procedure to the set of “correlated equilibrium” operating points, especially for the case of moving targets in the UGSN. Numerical results are presented in Section 4.4, and conclusions follow in Section 4.5. Appendix A summarizes the physical characteristics of the UGSN, including the ZigBee non-beacon-enabled communication protocol.

4.2 Decentralized Energy-Aware Node Activation

This section describes our decentralized approach to the sensor activation problem, which seeks to activate only the most well-positioned sensors, while the rest sleep. Section 4.2.1 formulates the model by which sensors obtain target measurements, Section 4.2.2 formulates a utility function to be maximized by each sensor node, Section 4.2.3 details the simple game theoretic adaptive algorithm that each sensor deploys to greedily optimize their utility, and its performance is analyzed in Section 4.2.4.

The task of the UGSN is to perform effective target detection and localization, using as few sensor nodes as possible so as to maximize lifetime. The problem of each sensor $l = 1, 2, \dots, L$ is to select an optimal action from the set:

$$\mathbf{S}^l = \{0 \text{ (sleep)}, 1 \text{ (activate)}\}. \quad (4.1)$$

We define the set \mathbf{S} to represent the joint action of all sensors. We consider a decentralized solution since it provides us with a simple and scalable algorithm for sensor node activation (since there is no central controller or lengthy negotiation), and since sleeping nodes do

not have to periodically turn on a receiver to obtain instructions, leading to greater energy efficiency.

4.2.1 Measurement Model

Sensor nodes use passive components to detect and relay information on targets with a specified sensor signature within their detection range. Let vector \mathbf{r}_n denote the set of target positions at discrete time n . Assuming the target signal intensity falls off as an inverse square of its range, the target signal level received by sensor node l is modelled as:

$$D^l(\mathbf{r}_n) = \min \left\{ \sum_{m=1}^M \frac{A_m}{(\Delta(l, r_{m,n}))^2} + \sigma_D(n), K_D \right\}, \quad (4.2)$$

where there are M targets, A_m represents the intensity of target $m = 1, 2, \dots, M$ at one metre, $\Delta(l, r_{m,n})$ is the distance from sensor node l to target m at discrete time n , K_D represents a signal saturation level, and $\sigma_D(n)$ is zero-mean, finite-variance and (possibly) spatially correlated noise at time n . Since we are concerned with sensor activation control, the measurement model is abstracted to a utility function in the next section. Because of this, the details of the measurement model are not important, and as a result we do not account for details such as the structure of the noise process.

4.2.2 Activation Control Utility Function

To implement decentralized activation control, each sensor l is equipped with an adaptive filter, which evaluates alternative actions $x^l \in \{1, 2\}$, denoting sleep and activate decisions for l , respectively. The evaluation criterion is a utility function $u^l(x^l, x^{-l})$, where x^{-l} represents the joint actions of the other sensors. The trade-off between target monitoring and energy usage is captured by setting:

$$u^l(x, \mathbf{r}) = \begin{cases} 0, & x^l = 0, \\ C^l(x, \mathbf{r}) - (E_{\text{packet}}(l, \eta^l(x)) + \Delta E_{\text{sens}}), & x^l = 1. \end{cases} \quad (4.3)$$

Here, $C^l(x, \mathbf{r})$ is a reward proportional to the expected contribution to an end user's utility associated with the information l can provide. The second two terms represent the energy cost associated with sensor activation and transmission efforts, and the details of their formulation can be found in Appendix A. The key feature that characterizes our study as a *game* is that these quantities are a function of x^{-l} , the number of other active nodes.

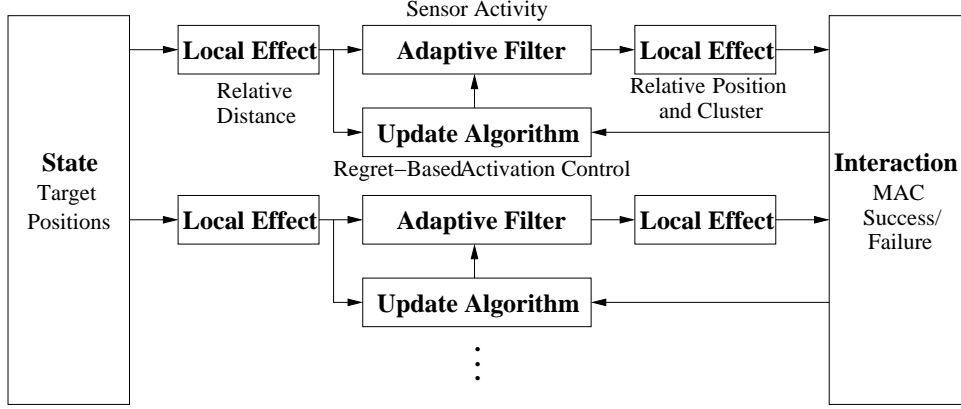


Figure 4.1: Decentralized adaptive filtering interpretation of sensor activation control. Sensors adapt to local target observations and inter-sensor competition through the ZigBee MAC layer.

To formulate the trade-off more carefully, let $\eta^l(x)$ and $\zeta^l(x)$ be given by:

$$\eta^l(x) = 1 + \sum_{k \in \mathfrak{C}_l(l)} x^k, \text{ and, } \zeta^l(x) = 1 + \sum_{k \in \mathfrak{N}_b(l)} x^k, \quad (4.4)$$

where $\mathfrak{C}_l(l)$ is the set of sensor nodes belonging to the same cluster as l , and $\mathfrak{N}_b(l)$ is the set of nodes “near” l (i.e. in neighboring clusters). η affects the success of transmission attempts, as in Section A.2, while ζ affects the contribution of a l ’s information, since other nearby active nodes are likely seeing the same target as l . Since the MAC layer performance is accounted for in the activation decision, the activation layer interacts with the ZigBee/IEEE 802.15.4 layer as in Figure 4.1.

The energy cost of transmission, E_{packet} , is specified in (A.4) in Appendix A as a function of l to allow transmission power to vary by sensor node. The additional energy cost of activating, not taken into account in (A.4), is given by ΔE_{sens} , computed from Table A.2 as follows:

$$\Delta E_{sens} = \frac{1}{F} [(1 - k_m)(P_{CPU} - P_{CPU_0} + P_S - P_{S_0}) - k_c P_{RX}(T_{CCA} + T_{RX_{wu}})], \quad (4.5)$$

where F is the frequency of sensor node activation decisions, and k_m, k_c are the (small) proportions of time spent sensing the environment and channel, respectively, in sleep mode. (It is assumed that sleeping nodes still monitor their environment at a low duty cycle.)

It remains to specify the expected reward function $C^l(\cdot)$. Assuming that information

increases with total signal strength, consider the well-known concave utility case for global (end-user) utility:

$$U_{glob}((y^l, y^{-l}), \mathbf{r}) = 1 - e^{-k_2 \min_m (\sum_j y^j(m))}, \quad (4.6)$$

where k_2 reflects the concavity of information and information value with signal strength, and $y^j(m)$ is the signal reading due to target m received by sensor j . If we assume that one target dominates the reading at any given sensor, we may write $y^j(m) = D^j s(\eta) x^j$, and l can approximate its marginal contribution for activating as:

$$\begin{aligned} C^l(x, \mathbf{r}) &= K_1 (U_{glob}((y^l, y^{-l}), \mathbf{r}) - U_{glob}((0, y^{-l}), \mathbf{r})) \\ &\approx K_1 \left(e^{-k_2 \bar{D}(\mathbf{r}) s(\eta) \zeta} \right) \left(1 - e^{-k_2 D^l(\mathbf{r}) s(\eta)} \right), \end{aligned} \quad (4.7)$$

where K_1 is a pricing parameter, and $\bar{D}(\mathbf{r})$ is the mean target signal intensity observed by other nearby sensors, which can be computed from the average sensor density and target characteristics. Values $s(\eta)$, η , and ζ are defined in Appendix A. To simplify matters, we have imposed here that each sensor assumes it is monitoring the most poorly observed target m , thus believing its contribution to U_{glob} is positive when it may in fact be zero. This avoids costly coordination of sensors monitoring targets in different regions, in keeping with our decentralized approach.

4.2.3 Nonlinear Adaptive Filtering for Sensor Activation

Having outlined the objectives of the UGSN, we now turn to our decentralized adaptive filtering approach to the sensor node activation. Our approach is based on the regret matching procedure of [133], formulated as a distributed stochastic approximation algorithm. This allows us to specify an adaptive variant of the original procedure, called “Regret Tracking,” which uses a constant stepsize to dynamically adapt to time varying conditions, so that sensor nodes may function in a changing environment. In the next section, we will show that the algorithm converges to the set of correlated equilibria of the decentralized sensor activation problem, posed in game theoretic terms.

The regret-tracking algorithm itself requires on-line evaluation of the utility (4.3), which in turn requires each sensor node to estimate $\hat{\eta}^l$ and $\hat{\zeta}^l$. In a decentralized system, this poses a problem, since the only information received by nodes is the success or failure of their previous channel access attempts. (See Appendix A for the Zigbee-based channel access architecture; sensors share channels through a “listen before talk” Carrier Sense Multiple Access scheme, wherein a series of “Clear Channel Assessment” (CCA) attempts

are performed by listening to channels in order to discover a usable one.)

Fortunately, the model of Section A.2 relates the proportion of successful CCA attempts to the traffic level through (A.1), which we can use to obtain maximum likelihood traffic estimates. Suppose sensor activity x remains relatively constant over the time it takes for a node to perform n CCA attempts. Then each CCA attempt represents a Bernoulli trial with success probability (probability of a clear channel) given by (A.1). It is straightforward to check that the maximum likelihood estimate of the traffic level $\eta^l(x)$ is given by the solution to:

$$\hat{\eta}^l(x) = \arg \max_{\eta} \left(\binom{n}{s} p(\eta)^s (1 - p(\eta))^{n-s} \right) = 1 + \log_{1-q} \left(\frac{s}{n} \right), \quad (4.8)$$

where s is the number of successful CCA attempts in the last n attempts, and the second equality is obtained via (A.1). A similar procedure can be used for estimating $\zeta^l(x)$. By tracking the proportion of successful CCA attempts, we may thus estimate activity through the following algorithm:

Algorithm 4.2.1 Estimating Active Sensors: Select $0 < \varepsilon_{act} < 1$. Select a history length $n > 0$. Repeat the following:

1. Each time a CCA is performed, update s as the number of successful CCAs of the last n attempts.
2. Compute $\hat{\eta}$ according to (4.8).
3. Periodically perform CCA attempts on the channel frequencies $\{f_1, \dots, f_k\}$ of any nearby clusters. Update s_k as the number of successful CCAs of the last n attempts.
4. Compute $\hat{\zeta} = \sum_{i=1}^k \hat{\zeta}_i$, where each $\hat{\zeta}_i$ is computed according to (4.8) with argument s_k .

Remark Since CCA successes are i.i.d., the estimates provided by Algorithm 4.2.1 are unbiased.

To ensure accurate estimates, we must ensure that CCAs are done with some minimum frequency, so that traffic levels do not change too frequently. Hence extra CCA activity must be carried out on top of those used for transmission. This is especially important when the sensor node is asleep for long periods. In sleep mode, this minimum frequency is set according to the quantity k_c in (4.5).

An advantage of this method is that very little signal processing needs to be done to determine whether a channel is busy or idle, which results in significant energy savings.

This can be carried even further; if sensors can directly sample the energy on the RF band, as suggested in [136] then the number of sensors can be estimated at a high duty cycle with very little energy expenditure.

The decentralized adaptive filtering, regret tracking algorithm itself is described as follows. At each decision period n , sensor nodes take joint action $X_n \in \mathbb{S}$, with Sensor l taking action $X_n^l \in \mathbf{S}^l$. To implement the algorithm, each filter/sensor l uses the potential utilities associated with past joint actions $\{X_n : n = 1, 2, \dots\}$ to derive regret values $\theta_{n,jk}^l : j, k \in \mathbf{S}^l$, according to:

$$\theta_{n,jk}^l = \sum_{\tau \leq n: X_\tau^l = j} \varepsilon_\tau \prod_{\sigma=\tau+1}^n (1 - \varepsilon_\sigma) \left(u^l((k, X_\tau^{-l}), \mathbf{r}_\tau) - u^l((j, X_\tau^{-l}), \mathbf{r}_\tau) \right). \quad (4.9)$$

If $\varepsilon_\tau = 1/(\tau + 1)$, this is simply the average:

$$\theta_{n,jk}^l = \frac{1}{n} \sum_{\tau \leq n: X_\tau^l = j} \left(u^l((k, X_\tau^{-l}), \mathbf{r}_\tau) - u^l((j, X_\tau^{-l}), \mathbf{r}_\tau) \right).$$

If $\varepsilon_\tau = \varepsilon$, it is the exponentially weighted moving average:

$$\theta_{n,jk}^l = \sum_{\tau \leq n: X_\tau^l = j} \varepsilon (1 - \varepsilon)^{n-\tau} \left(u^l((k, X_\tau^{-l}), \mathbf{r}_\tau) - u^l((j, X_\tau^{-l}), \mathbf{r}_\tau) \right).$$

These values reflect the (weighted) average potential difference in utility had l used action k instead of action j in the past. Node l then switches from its current action j to k with probability proportional to $\theta_{n,jk}^l$ (if it is non-negative).

The complete, decentralized procedure is summarized below:

Algorithm 4.2.2 Nonlinear Adaptive Filter for ZigBee Node Activation: The regret-based algorithm for sensor node activation has parameters $(u^l, \{\varepsilon_n : n = 1, 2, \dots\}, \theta_0^l, X_0^l)$, where u^l are the utilities, $\{\varepsilon_n\}$ is a small stepsize, and θ_0^l, X_0^l are arbitrary initial regrets and actions.

Define the $\mathbf{S}^l \times \mathbf{S}^l$ matrix with entries:

$$\mathbf{H}_{jk}^l(X_n, \mathbf{r}_n) = I\{X_n^l = j\} \left(u^l((k, X_n^{-l}), \mathbf{r}_n) - u^l((j, X_n^{-l}), \mathbf{r}_n) \right). \quad (4.10)$$

Execute the following:

1. **Initialization:** Set $n = 0$, and take action X_0^l . Initialize regret $\theta_0^l = \mathbf{H}^l(X_0, \mathbf{r}_0)$.

2. Repeat for $n = 0, 1, 2, \dots$, the following steps:

Action Update: Choose $X_{n+1}^l = k$ with probability

$$Pr(X_{n+1}^l = k | X_n^l = j, \theta_n^l = \theta^l) = \begin{cases} \max\{\theta_{jk}^l, 0\}/\mu, & k \neq j, \\ 1 - \sum_{i \neq j} \max\{\theta_{ji}^l, 0\}/\mu, & k = j, \end{cases} \quad (4.11)$$

where $\mu > \sum_{k \neq j} \max\{\theta_{jk}^l, 0\}$.

Regret Value Update: Estimate $(\hat{\eta}_{n+1}, \hat{\zeta}_{n+1})$, and compute utility $u^l(0, X_{n+1}^{-l})$ and $u^l(1, X_{n+1}^{-l})$, hence $\mathbf{H}^l(X_{n+1})$. Update θ_{n+1} using the stochastic approximation (SA):

$$\theta_{n+1}^l = \theta_n^l + \varepsilon_n(\mathbf{H}^l(X_{n+1}, \mathbf{r}_{n+1}) - \theta_n^l), \quad (4.12)$$

Note that θ_n^l is a moving average of the updates $\{H^l(X_k, r_k) : k = 1, 2, \dots, n\}$. Because of this, Algorithm 4.2.2 can be viewed as a stochastic approximation algorithm with a constant stepsize $\varepsilon > 0$; actions are chosen with probability proportional to their moving average potential performance in the past. For the original regret matching algorithm of [133], one would instead use $\varepsilon_n = 1/(n+1)$. We also point out the generality of Algorithm 4.2.2, by noting that it can be easily transformed into the well-known fictitious play algorithm (see [130]) by choosing $X_{n+1}^l = \arg \max_k \theta_{jk}^l$ deterministically, where $X_n^l = j$, and the extremely simple best response algorithm by further specifying $\varepsilon = 1$. However, in simulation, these two variants do not perform as well as the regret tracking scheme (see Section 4.4). Finally, we use a variable normalization μ as in [126], as opposed to the static μ of [133]. This greatly improves the speed of convergence to equilibrium.

The target position \mathbf{r}_n is regarded as a slowly varying parameter of the utility function in Algorithm 4.2.2. Since the utility varies (due to moving targets), a constant stepsize in Algorithm 4.2.2 is needed to keep sensor nodes responsive to the changes.

At each iteration, Algorithm 4.2.2 requires estimates of the activity of other nodes to compute $u^l(1, X_n^{-l})$. This is carried out through Algorithm 4.2.1. Other than this, the computational complexity of Algorithm 4.2.2 is small, and therefore suitable for implementation in a UGSN. At each iteration, each filter/sensor needs to perform one table lookup to compute the utility given the current readings of (X_n^{-l}, \mathbf{r}_n) . Two additions and two multiplications update the regret value, and one random number, one multiplication and one comparison suffices to compute the next action. In total, assuming random numbers are stored in a table, this yields two table lookups, one comparison, and five arithmetic operations per iteration.

4.2.4 Performance of Regret Tracking

Our main result is that if each adaptive filter operates according to Algorithm 4.2.2, the system converges to the set of correlated equilibria. This provides a performance guarantee for the sensor network: each sensor converges to its best action given its limited information and coordination capabilities. We will investigate the meaning and details of this convergence in the next section. Concerning the overall performance of the correlated equilibrium, the following two results are established:

Theorem 4.2.1 *1. If the correlated equilibrium is deterministic, and the number of sensors in each cluster is less than the critical point of (A.4), then the resulting sensor deployment is a Pareto optimal Nash equilibrium.*

2. If the correlated equilibrium is deterministic, and the overall system objective is to maximize:

$$K_1 U_{glob}(y, \mathbf{r}) - \sum_l x^l (E_{packet}(l, \eta^l(x)) + \Delta E_{sens}) \quad (4.13)$$

(i.e. trade off the end-user utility with the total energy consumed, see (4.5), (4.6) (A.4)) then the equilibrium deployment is optimal in the sense that any unilateral deviation will reduce (4.13).

The condition in Part 1 of Theorem 4.2.1 essentially guarantees that the utility (4.3) is concave in the number of active sensors. This implies that no group of sensors can improve their situation cooperatively if they cannot do so independently. A formal proof follows.

Proof For the first part, we wish to show that there is no other deployment that makes at least one sensor better off and no sensor worse off (a deterministic correlated equilibrium is a Nash equilibrium by definition). Let the equilibrium point be X and define $\tilde{X}^l = 0$ when $X^l = 1$ and vice-versa. Consider a sensor l such that $X^l = 1$. By the equivalence of negative regret and correlated equilibrium set (see Section 4.3), we have $\theta_{n,10}^l = u^l((0, X_n^{-l}), \mathbf{r}_n) - u^l((1, X_n^{-l}), \mathbf{r}_n) \leq 0$, i.e. $u^l((1, X_n^{-l}), \mathbf{r}_n) > 0$, which implies that l is worse off if it chooses action 0 (and gets zero reward), regardless of others' actions. For a sensor $k : X^k = 0$, we have $\theta_{n,01}^k = u^k((1, X_n^{-k}), \mathbf{r}_n) - u^k((0, X_n^{-k}), \mathbf{r}_n) \leq 0$, i.e. $u^k((1, X_n^{-k}), \mathbf{r}_n) < 0$, so k cannot gain by unilaterally choosing $X^k = 1$. Moreover, if multiple sleeping sensors simultaneously activate, their loss will be even worse, since $C^l(x, \mathbf{r})$ is decreasing in x , and $E_{packet}(x)$ is increasing by hypothesis. Hence there is no globally beneficial deviation and Pareto optimality holds. For the second part, it is enough to note that $u^l((1, X_n^{-l}), \mathbf{r}) \geq 0$ iff $K_1 U_{glob}((y^l, y^{-l}), \mathbf{r}) - (E_{packet}(l, \eta^l(x)) + \Delta E_{sens}) \geq K_1 U_{glob}((0, y^{-l}), \mathbf{r}) - 0$. Since each

sensor activates if and only if its utility is positive, a consideration of (4.13) shows the result holds.

The analysis above reveals a connection between optimality and the set of correlated equilibria, which results from our choice of utility function. This suggests that our approach is competitive with similar approaches to sensor deployment, either centralized or decentralized.

4.3 Correlated Equilibrium and Convergence

This section deals with the convergence and tracking properties of Algorithm 4.2.2. Section 4.3.1 defines the correlated equilibrium set, and discusses its interest in decentralized systems. Sections 4.3.2 and 4.3.3 present technical convergence and tracking results for Algorithm 4.2.2 via a differential inclusion approach. Section 4.3.4 presents improvements for Algorithm 4.2.2 through pricing and outer-level stochastic optimization.

The set of correlated equilibria to which Algorithm 4.2.2 converges (shown below) is a convex set of strategies representing competitively optimal behaviour between sensors. Theoretically, finding a correlated equilibrium in a game is equivalent to solving a linear feasibility problem, which is a linear program with a null objective function. Algorithm 4.2.2 is essentially an iterative procedure for solving this problem, which is interesting because of its decentralized nature.

4.3.1 Definition of Correlated Equilibrium

For a game with L players, the problem is for each player $l = 1, 2, \dots, L$ to devise a rule for selecting actions x^l from a set \mathbf{S}^l (with size S^l), to maximize the expected value of their utility function $u^l(x^1, x^2, \dots, x^L)$. (To maintain generality, we suppress the dependence of u^l on \mathbf{r} in this section.) Since each player only controls one of L variables, this requires careful consideration of the expected actions of other players. The standard solution to this problem is an equilibrium, which identifies stable operating points of the game. The most common such equilibrium is due to Nash [146]. However, we focus on an important generalization of the Nash equilibrium, proposed in [124, 125], known as the *correlated equilibrium*, defined as follows:

Definition 4.3.1 *Define a joint strategy π to be a probability distribution on the product space $\mathbb{S} = \mathbf{S}^1 \times \mathbf{S}^2 \times \dots \times \mathbf{S}^L$. Any π may be decomposed into marginals (π^l, π^{-l}) for any*

l , where π^l is the marginal probability distribution (strategy) of l , and π^{-l} is the marginal distribution of all players but l . Now, π is a correlated equilibrium if

$$\sum_{x^{-l} \in \mathbf{S}^{-l}} \pi(j, x^{-l}) [u^l(k, x^{-l}) - u^l(j, x^{-l})] \leq 0. \quad (4.14)$$

One intuitive interpretation is that π provides each player l with a private “recommendation” $a^l \in \mathbf{S}^l$. Based on this, and knowing π , a player could compute *a posteriori* probability distribution for the actions of other players, and hence an expected utility for each action. The equilibrium condition states that there is no deviation rule that would award l a better expected utility.

Assuming \mathbf{S}^l is the same for all l , condition (4.14) can be written in matrix form as: $A\pi \leq 0$, $\mathbf{1}\pi = 1$, where $\mathbf{1}$ represents a row vector of ones, and A is an $L(\mathbf{S}^l)^2 \times (\mathbf{S}^l)^L$ matrix with entries

$$A_{jk, (i, m)}^l I\{i = j\} (u^l(k, m) - u^l(j, m)), \quad (4.15)$$

where l, j, k correspond to a unique row of A , and (i, m) correspond to a unique column. There is a distinct similarity between (4.15) and (4.10); if we let $\pi(X_n)$ denote the representation of joint action X_n in the space of probability distributions on \mathbb{S} , then the regret update equation (4.12) in Algorithm 4.2.2 can be rewritten (simultaneously for all players) as:

$$\theta_{n+1} = \theta_n^l + \varepsilon_n (A(\pi(X_{n+1})) - \theta_n). \quad (4.16)$$

From (4.15), it is clear that the correlated equilibria comprise a convex set, and we denote by \mathcal{C}_e the set of $\pi \in \Delta\mathbb{S}$ satisfying (4.14) for all l, j, k .

There are several benefits for considering a correlated equilibrium over a Nash. The correlated equilibrium concept permits coordination between players, and can lead to improved performance over a Nash equilibrium [125]. The set (4.14) is also structurally simpler than the set of Nash equilibria; it is a convex set, whereas the Nash equilibria are isolated points at the extrema of this set [147]. Since the set of correlated equilibria is convex, fairness between players can also be addressed in this domain. Finally, decentralized, online adaptive procedures naturally converge to (4.14), whereas the same is not true for Nash equilibria (the so-called law of conservation of coordination [135]).

4.3.2 Convergence Analysis of Algorithm 4.2.2

Our goal is to show that when each filter/sensor deploys Algorithm 4.2.2, the global network performance of the UGSN (that is, the empirical distribution of play below) for the UGSN converges to the set of correlated equilibria (4.14).

Definition 4.3.2 *Define the global network performance (empirical distribution of play) up to time n by*

$$\bar{z}_n = \sum_{\tau \leq n} \varepsilon_\tau \left(\prod_{\sigma=\tau+1}^n (1 - \varepsilon_\sigma) \right) e_{X_\tau}, \quad (4.17)$$

where $e_x = [0, 0, \dots, 1, 0, \dots, 0]$ with the one in the x^{th} position.

Here ε_n is a weighting factor. This yields

$$\bar{z}_n = \frac{1}{n} \sum_{\tau} e_{X_\tau}, \text{ or, } \bar{z}_n = \varepsilon \sum_{\tau} (1 - \varepsilon)^{n-\tau} e_{X_\tau}, \quad (4.18)$$

depending on whether $\varepsilon_n = 1/(n+1)$ or $\varepsilon_n \equiv \varepsilon > 0$, respectively. Note that in both cases \bar{z}_n is an empirical frequency, since $\sum_i \bar{z}_n(i) = 1$.

We are interested here, as in [133], in convergence to the set of correlated equilibria (\mathcal{C}_e). Recall that $\bar{z}_n \rightarrow \mathcal{C}_e$ as $n \rightarrow \infty$ if

$$\text{dist}(\bar{z}_n, \mathcal{C}_e) = \inf_{y \in \mathcal{C}_e} |\bar{z}_n - y| \rightarrow 0 \text{ as } n \rightarrow \infty. \quad (4.19)$$

Because of the set theoretic nature of the results, analysis is carried out via a differential inclusion, the set theoretic extension of a differential equation.

In order to present the convergence of \bar{z}_n to the set of correlated equilibria, it is convenient to represent the global network performance \bar{z}_n in (4.17) by the following recursion:

$$\bar{z}_{n+1} = \bar{z}_n + \varepsilon_n (e_{X_{n+1}} - \bar{z}_n), \quad (4.20)$$

where $\{\varepsilon_n\}$ is either a sequence of decreasing stepsize or a small constant $\varepsilon_n = \varepsilon > 0$, and X_{n+1} is constructed according to Step (2) of Algorithm 4.2.2 and $\{\varepsilon_n\}$ is a sequence of stepsizes. We will use both $\varepsilon_n = 1/(n+1)$ and $\varepsilon_n = \varepsilon$ (a small constant stepsize). As is widely used in stochastic approximation proofs [140], we also introduce the following continuous time interpolated processes.

Definition 4.3.3 For a constant stepsize $\varepsilon > 0$, define the interpolated global network performance

$$\bar{z}^\varepsilon(t) = \bar{z}_n \text{ for } t \in [\varepsilon n, \varepsilon(n+1)). \quad (4.21)$$

For a decreasing stepsize sequence $\varepsilon_n = 1/(n+1)$ is used, we define

$$t_n = \sum_{j=0}^{n-1} \varepsilon_j, \quad m(t) = \max\{n : t_n \leq t\}.$$

Define a sequence of piecewise constant interpolations as $\bar{z}^0(t) = \bar{z}_n$ for $t \in [t_n, t_{n+1})$ and define the shift sequence by

$$\bar{z}^n(t) = \bar{z}^0(t + t_n).$$

We present two lemmas, the proof of the first lemma can be found in [140, Chapters 6 and 8]. This lemma (a generic differential inclusion result for any x_n process) will be used to obtain the convergence results. Recall that a generic differential inclusion is of the form $dz/dt \in G(z)$, i.e., it specifies a family of trajectories. It is a generalization of a differential equation $dz/dt = g(z)$ which comprises of a single trajectory.

Lemma 4.3.1 *Consider a generic stochastic approximation algorithm $x_{n+1} = x_n + \varepsilon_n g(x_n, \zeta_n)$, where ε_n is either a constant stepsize $\varepsilon_n = \varepsilon$ or $\varepsilon_n = 1/(n+1)$. Assume that for each ζ , $g(\cdot, \zeta)$ is continuous, and for each x , $g(x, \cdot)$ is bounded. Moreover, assume that for each fixed x , there is a set $G(x)$ such that*

$$\text{dist} \left[\frac{1}{n} \sum_{j=m}^{n+m-1} g(x, \zeta_j), G(x) \right] \rightarrow 0 \begin{cases} \text{in probability when } \varepsilon_n = \varepsilon, \\ \text{almost surely when } \varepsilon_n = 1/(n+1), \end{cases}$$

Define the interpolated process $x^n(\cdot)$ and $x^\varepsilon(\cdot)$ as in Definition 4.3.3. Then $x^n(\cdot)$ and $x^\varepsilon(\cdot)$ converges to $x(\cdot)$ almost surely (for the decreasing stepsize case) and weakly (for the constant stepsize case), where the limit $x(\cdot)$ is a solution of $\dot{x} \in G(x)$.

Define the following differential inclusion as in [126]

$$\frac{dz}{dt} \in \nu(z) \times \Delta S^{-l} - z, \quad (4.22)$$

where $\nu(z)$ is the set of probability distributions over l 's action set \mathbf{S}^l , such that

$$\sum_k \nu_k [\bar{\mathbf{H}}_{kj}^l(z)]^+ = \nu_j \sum_k [\bar{\mathbf{H}}_{jk}^l(z)]^+$$

for all $j, k \in \mathbf{S}^l$. Here ΔS^{-l} is the set of all probability distributions over joint actions of the remaining $L - 1$ sensor nodes, and

$$\bar{\mathbf{H}}_{jk}^l(z) = \sum_{x_i \in \mathbf{S}^k} \mathbf{H}^l(x_i) z(i). \quad (4.23)$$

Convergence of (4.22) to the set of correlated equilibria (4.14) is established in Section 5 of [126] by a Liapunov argument. Let us now present the convergence of the algorithm.

Theorem 4.3.1 *Consider the interpolated processes defined in (4.3.3) generated by Algorithm 4.2.2. The following results hold:*

1. *Under a decreasing stepsize $\varepsilon_n = 1/(n + 1)$ in Algorithm 4.2.2, the trajectory of the interpolated SA $\bar{z}^n(\cdot)$ for the global network performance converges almost surely to a trajectory $z(\cdot)$ whose dynamics are given by (4.22).*
2. *For the decreasing stepsize algorithm, as $n \rightarrow \infty$, for any $q_n \rightarrow \infty$ as $n \rightarrow \infty$, $\bar{z}^n(\cdot + q_n)$ converges almost surely to the set of correlated equilibria.*
3. *Under a constant stepsize $\varepsilon_n = \varepsilon$ in Algorithm 4.2.2, $\bar{z}^\varepsilon(\cdot)$ converges weakly as $\varepsilon \rightarrow 0$ to a trajectory $z(\cdot)$ that is given by (4.22).*
4. *For the constant stepsize algorithm, as $\varepsilon \rightarrow 0$, for any $q_\varepsilon \rightarrow \infty$ as $\varepsilon \rightarrow 0$, $\bar{z}^\varepsilon(\cdot + q_\varepsilon)$ converges weakly to the set of correlated equilibria.*

Outline of Proof: First, note that by the argument of [126, see pp. 334-335], it can be shown that

$$\text{dist} \left[\frac{1}{n} \sum_{j=m}^{n+m-1} e_{X_j}, \mathcal{C}_e \right] \rightarrow 0 \quad (4.24)$$

as $n \rightarrow \infty$ for any $m \geq 0$, where the convergence takes place either in the sense of almost surely for the decreasing stepsize case, or in probability for the constant stepsize case. With (4.24) holding, Lemma 4.3.1 implies that the interpolated processes defined in Definition 4.3.3 converges to $z(\cdot)$ (either almost surely or weakly) that satisfies (4.22). The rest of the proof follows from the standard argument as in [140]. The details are omitted.

For static game conditions, and stepsize $\varepsilon_n = 1/(n + 1)$, Algorithm 4.2.2 is exactly the regret matching algorithm introduced in [133], and convergence to the set of correlated equilibrium is proven there. For dynamic game conditions, Theorem 4.3.1 states that if the game varies slow enough, then the equilibrium set is still tracked through time by the

algorithm. This is important because it guarantees that the pricing modifications in Section 4.3.4 are valid, and that slowly moving targets and sensor failures are supported.

4.3.3 Tracking Performance of Algorithm 4.2.2

The above analysis assumes that the target does not move. The purpose of this section is to analyze the tracking performance of Algorithm 4.2.2 when the target process $\{\mathbf{r}_n : n = 0, 1, 2, \dots\}$ evolves slowly according to a finite state Markov chain. Given that the measurement process at sensors occurs at a much faster time scale compared to the movement of the targets, formulating the target dynamics as a slow Markov chain is a reasonable assumption. We show that Algorithm 4.2.2 can satisfactorily track the set of correlated equilibria of the network. It is important to keep in mind that the Markovian assumption of the target is only used in our analysis below. The implementation of Algorithm 4.2.2 does not require the dynamics of the target to be known explicitly.

Suppose, for the purpose of analysis, \mathbf{r}_n , the target position parameterizing (4.10), evolves according to a slow Markov chain with a finite state space $\mathcal{M}_r = \{\mathbf{r}^1, \dots, \mathbf{r}^m\}$ and a transition matrix of the form $P = P^\rho = I + \rho Q$, where I is the identity matrix, $\rho > 0$ is a small parameter, and Q is a generator of a continuous-time Markov chain (i.e., $Q = (q_{ij})$ satisfying $q_{ij} \geq 0$ for $i \neq j$, $\sum_j q_{ij} = 0$ for each i). In what follows, we work with $\rho = \mathcal{O}(\varepsilon)$ or $\rho = \mathcal{O}(\varepsilon^2)$. Clearly if $\rho = \mathcal{O}(\varepsilon^2)$, then the target evolves on a slower time scale compared to the adaptation speed of the stochastic approximation Algorithm 4.2.2 which has step size ε . On the other hand if $\rho = \mathcal{O}(\varepsilon)$, then the target evolves on the same time scale as the adaptation speed of Algorithm 4.2.2. Depending on the time scale in which the target evolves, the following theorem analyses the tracking performance of Algorithm 4.2.2.

Define the interpolated process $\bar{z}^\varepsilon(\cdot)$ as in the previous section and define $\mathbf{r}^\varepsilon(t) = \mathbf{r}_n$ for $t \in [n\varepsilon, n\varepsilon + \varepsilon)$.

Theorem 4.3.2 *Using a constant stepsize $\varepsilon_n = \varepsilon$ in Algorithm 4.2.2, the following hold:*

1. Mean Square Tracking Error for Slow Target: *If $\rho = \mathcal{O}(\varepsilon^2)$, then for each $y \in \mathcal{C}_e$ and for n sufficiently large, $E|\bar{z}_n - y|^2 = \mathcal{O}(\varepsilon)$.*
2. Differential Inclusion Limit for Slow Target: *If $\rho = \mathcal{O}(\varepsilon^2)$, then as $\varepsilon \rightarrow 0$, the pair of processes $(\bar{z}^\varepsilon(\cdot), \mathbf{r}^\varepsilon(\cdot))$ converges weakly to $(z(\cdot), \mathbf{r}(\cdot))$ where $(z(\cdot), \mathbf{r}(\cdot))$ satisfy the differential inclusion*

$$\frac{dz}{dt} \in \nu(z) \times \Delta S^{-l} - z, \text{ with} \quad (4.25)$$

$$\sum_k \nu_k [\bar{\mathbf{H}}_{kj}^l(z)]^+ = \nu_j \sum_k [\bar{\mathbf{H}}_{jk}^l(z)]^+.$$

Here \mathbf{r}_n is the actual state of the target process, $\bar{z}^\varepsilon(\cdot)$ is defined in (4.21), and $\bar{\mathbf{H}}(\cdot)$ is defined analogously to (4.23), but with explicit dependence on \mathbf{r}_n .

3. **Differential Inclusion Limit for Fast Target:** If $\rho = \mathcal{O}(\varepsilon)$, then as $\varepsilon \rightarrow 0$, $(\bar{z}^\varepsilon(\cdot), \mathbf{r}^\varepsilon(\cdot))$ converges weakly to $(z(\cdot), \mathbf{r}(\cdot))$ where $(z(\cdot), \mathbf{r}(\cdot))$ satisfy the Markovian switched differential inclusion (4.25) with

$$\sum_k \nu_k [\bar{\mathbf{H}}_{kj}^l(z, r(t))]^+ = \nu_j \sum_k [\bar{\mathbf{H}}_{jk}^l(z, r(t))]^+$$

Here $\mathbf{r}(t)$ is a continuous-time Markov process with generator Q .

Remark Due to limited space, we provide an outline of the proof. The proof is a combination of convergence analysis provided in the last section and the two-time-scale formulation for Markov chains (see [153]). By a slow Markov chain, we mean that the chain changes its state relatively infrequently. Thus the time-varying parameter stays in a state (a constant) at most of the time, and jumps to a new location at random instance. Note that the infrequent changes make it possible for us to track the time-varying parameter process. If it changes too fast, then there is virtually not possible to keep track of the time-varying parameter. This is known as trackability. In the traditional analysis, one often works with a time-varying parameter whose trajectories change slowly but continuously. Here, we allow the parameter jump changes with possibly large jump size. However, we require the jumps do not happen too frequently.

The results depend on how “slow” the Markov chain changes. If the dynamics of the Markov chain, namely, the transition of the Markov chain changes much slower than the adaptation rate, then the effect of this variation can be asymptotically “ignored.” The detailed analysis for this claim follows from the analysis given in [152]. On the other hand, if the Markov chain dynamics evolve in the same order of magnitude as that of the adaptation speed, then a Markov regime-switched mean dynamic system is obtained.

For the proof of part 1, using a perturbed Liapunov function, argument [140] (see [152] for details), for small ε and ρ , we obtain for n sufficiently large, $E|\bar{z}_n - y|^2 = \mathcal{O}(\rho + \varepsilon + \varepsilon^2/\rho)$. Note that to obtain such a result, we have not used any relative order of ε and ρ up to this point. Next, using $\rho = \varepsilon^2$, we establish the desired result. As a consequence, $E[\text{dist}(\bar{z}_n, \mathcal{C}_e)^2] = \mathcal{O}(\varepsilon)$. The rest of the proof is a modification of two-time-scale algorithms

for tracking parameter variations studied in previous work [152, 154] in conjunction with the differential inclusion for correlated equilibria. In what follows, we only indicate the main steps of the proof. In fact, we will only comment on the last assertion. Recall that in this case, $\rho = \mathcal{O}(\varepsilon)$. For convenience, we simply put $\rho = \varepsilon$ in what follows. This simplification is adopted mainly because this part of the proof is more difficult than the first two assertions. Denote by $p_n^\varepsilon = (P(\mathbf{r}_n^\varepsilon = \mathbf{r}^1), \dots, P(\mathbf{r}_n^\varepsilon = \mathbf{r}^m))' \in \mathbb{R}^m$, the probability distribution vector. Then it can be show that for each $T > 0$ and $0 \leq n \leq T/\varepsilon$, $p_n^\varepsilon = p(t) + \mathcal{O}(\varepsilon + \exp(-\kappa_0 t/\varepsilon))$, with $p(t)$ satisfying $\dot{p}(t) = Q'p(t)$. In the above, for $X \in \mathbb{R}^{\iota_1 \times \iota_2}$ with $\iota_1, \iota_2 \geq 1$, X' denotes its transpose. Similarly, we obtain related convergence result for the corresponding transition probability matrix. Next, we can show that $\mathbf{r}^\varepsilon(\cdot)$ converges weakly to a Markov chain whose generator is given by Q ; see [153] (also [154]) for details. To proceed, consider the pair $(\bar{z}^\varepsilon(\cdot), \mathbf{r}^\varepsilon(\cdot))$. We use martingale averaging ideas to obtain the desired limit. To do so, we show that the two-component process $(\bar{z}^\varepsilon(\cdot), \mathbf{r}^\varepsilon(\cdot))$ is tight in the space of functions that are right continuous with left limits endowed with the Skorohod topology (see [140]). Then we can extract weakly convergent subsequence. For simplicity, still index the subsequence by ε with limit denoted by $(z(\cdot), \mathbf{r}(\cdot))$. We then use the averaging techniques to finish up the proof (see [154]).

4.3.4 Algorithm Improvements and Pricing

Under Algorithm 4.2.2, if target activity persists in a given area for a long period, the adaptive filters will tend to converge to a single action, tending to activate some nearby sensors to monitor the target indefinitely. This may waste resources, prematurely burning out sensors instead of sharing the load. Another drawback is that regret averaging results in a lag between target motion and an appropriate response. To remedy these problems, we introduce two improvements: a two-level pricing dynamic, which changes sensor incentives if they are steadily active for a specified number of iterations, and a reset mechanism for the regret, which activates in the event that no targets are nearby.

The two improvements can be implemented via the following algorithm3:

Algorithm 4.3.1 Timer-Based Pricing/Event-Based Reset: Each sensor is set with timeout parameter ϕ , and pricing levels $K_1(\text{Normal}) > K_1(\text{Backoff})$ and performs the following:

1. **Initialization:** Initialize an action counter $\text{cnt} = 0$, and define a “Normal” and “Backoff” state. Initially set “State”=“Normal.” Also set a threshold parameter D_{\min} for the sensed target distance.

2. Repeat for iteration $n = 1, 2, \dots$:
 - if $X_n^l = X_{n-1}^l$, i.e. if the action from the last iteration is repeated, increment counter cnt .
3. if $cnt = \phi$, toggle the state between “Normal” and “Backoff,” and reset $cnt = 0$.
4. Set $K_1 = K_1(\text{state})$ in (4.7) for utility calculations in Algorithm 4.2.2.
5. Concurrently, reset the regret values according to sensor readings in either of the following events:
 - (a) If $X_n^l = 1$ and $D^l < D_{\min}$, reset the filter regret $\theta_n^l(1, 0) = 1/\varepsilon$.
 - (b) If $X_n^l = 0$ and $D^l < D_{\min}$, reset $\theta_n^l(0, 1) = -\varepsilon$.

Algorithm 4.3.1 introduces a two-level pricing dynamic which tends to periodically switch the sensor on or off (if $K_1(\text{Backoff})$ is low enough) periodically, allowing other nearby sensors to rest or take over as appropriate. It also mitigates the adverse effects of using averaging to track the regret values in two ways. First, if an active sensor suddenly loses track of its target(s), it will be endowed with a high regret for remaining active and thus quickly switch to sleep mode. Second, if an inactive sensor suddenly observes a target, it will be prepared to switch quickly to active mode should a target appear since its average regret is only slightly negative.

Provided that Algorithm 4.2.2 utilizes a constant stepsize, the changes initiated by Algorithm 4.3.1 can be tracked by the sensors, and the result can significantly improve sensor network operation. Timer parameters must be chosen carefully; faster timeouts distribute effort more even, but also impact performance since the best sensors are often in backoff. We study the trade-off through simulation in Section 4.4.

The pricing levels (and similarly the timer parameter ϕ) can be set by further stochastic approximation on a slow time scale relative to that of Algorithm 4.2.2. To motivate, suppose the end user divides time into long decision periods labeled $m = 1, 2, \dots$. At the end of each period m , the unpriced global utility,

$$U(\bar{E}) = 1 - e^{-k_2 \sum_j y^j}, \quad (4.26)$$

(see (4.6)) received in period m can be computed, along with the total energy \bar{E} spent by all sensors. Assuming the end user discounts future utilities by $\alpha < 1$, and that the present value of $U(\bar{E})$ is indicative of future values, the end user should set the relative sensor prices

(parameter $K = K_1(\text{Normal})/K_1(\text{Backoff})$ in Algorithm 4.3.1 to maximize:

$$U_{total} = \sum_{m=1}^{N(\bar{E})} U(\bar{E})\alpha^m = U(\bar{E})\frac{\alpha}{1-\alpha} \left(1 - \alpha^{N(\bar{E})}\right), \quad (4.27)$$

(where $N(\bar{E}) \approx E_{net}/\bar{E}$ is the network lifetime; E_{net} is the total energy remaining in the network.) Thus, the sensor utilities are periodically varied by a broadcast transmission from the base station, which may use a stochastic gradient algorithm (e.g. Kiefer-Wolfowitz) to select the current relative price K . The base station estimates $U(\bar{E}), N(\bar{E})$ based on the number and content of reports received and evaluates (4.27). Then derivative dU_{total}/dK is estimated using a finite difference or simultaneous perturbation methods as in [148], and a new pricing parameter K is broadcast to be used by each sensor in interval m , satisfying:

$$K(m+1) = K(m) + \beta_m \frac{dU_{total}}{dK}(K(m)), \quad (4.28)$$

for stepsize β_m .

The following is a standard result from stochastic optimization (see [140, 148]):

Theorem 4.3.3 *The sequence $K(m) : m = 1, 2, \dots$ computed in (4.28), with unbiased estimates for $dU_{total}(K(m))/dK$, converges almost surely to a locally optimal price, provided that $\beta_m > 0$, $\beta_m \rightarrow 0$, $\sum_{m=1}^{\infty} \beta_m = \infty$, and $\sum_{m=1}^{\infty} \beta_m^2 < \infty$.*

The slow time scale stochastic pricing operations described above are important factors in the long-term performance of the UGSN. There are many other parameters that could benefit from this approach, including the stepsize in Algorithm 4.2.2, the various utility parameters, and even the ZigBee operational characteristics. Providing a unified framework for treating all these parameters as prices to be optimized would be an interesting exercise, but beyond the scope of this paper.

4.4 Test Bed and Numerical Results for ZigBee-enabled UGSN

In this section we present simulation results illustrating the algorithms in Sections 4.2.3 for representative scenarios. For this purpose, a simulation test bed for ZigBee sensor nodes was developed in `Matlab`, using the `TrueTime` simulation package, [123]. This package provides a ready-made event-based simulator for ZigBee compatible UGSNs, which can be

Table 4.1: Simulation Parameters

Parameter	Value	Description
K_1	20	Pricing parameter, Eq. (4.7)
k_2	3	Utility concavity, Eq. (4.7)
\bar{D}	0.1	Mean signal strength, Eq. (4.7)
F	20 <i>Hz</i>	Decision frequency, q , Eq. (4.5)
D	100 bytes	Data block size, q , Eq. (A.4)
k_m	0.01	Target monitoring activity (sleeping), Eq. (4.5)
k_c	0.01	Channel monitoring activity (sleeping), Eq. (4.5)
A_m	1	Target signal intensity, Eq. (4.2)
K_D	1	Sensor saturation, Eq. (4.2)
ε_{act}	0.2	Channel averaging memory, Algorithm 4.2.1
ε	0.05	Regret matching memory, Algorithm 4.2.2

run in parallel with other `Matlab` code, allowing us to simultaneously simulate both target dynamics and node activity. The node activity in turn comprises of Algorithms A.2.1, 4.2.1, and 4.2.2. Note Algorithm A.2.1 is fully specified by the `TrueTime` package.

The adaptive filtering algorithm 4.2.2 specifies that sensor nodes should activate only when they are close enough to a target, and far enough from other active nodes. In this section we discuss the performance of the regret tracking algorithm for the following scenario:

- The UGSN comprises of $L = 500$ sensor nodes deployed uniformly over a 100×100 meter grid, and 10 hubs.
- We consider $M = 1$ and $M = 10$ targets, each of which radiates an acoustic signature with mean intensity 1 and variance 0.05, at one metre.

We consider a bandwidth-intensive application, in which activation decisions are made every 20 ms, and 100 bytes of data are transmitted each decision period by active sensor nodes. The parameter values for the simulation are specified in Table 4.1. Physical parameters are $k_m = k_c = 0.01$, $K_D = 4$, meaning that sleeping sensors only wake up 1% of the time for monitoring, and that the sensor itself is saturated when its target is closer than 0.5 metre. The value $\bar{D} = 0.1$ is computed by considering the average distance from a target to a sensor, given 500 sensors in a 100×100 metre grid.

The adaptive filtering Algorithm 4.2.2 was embedded into the simulated ZigBee test bed, and the outcomes averaged over 30 independent simulations were used to obtain the results below.

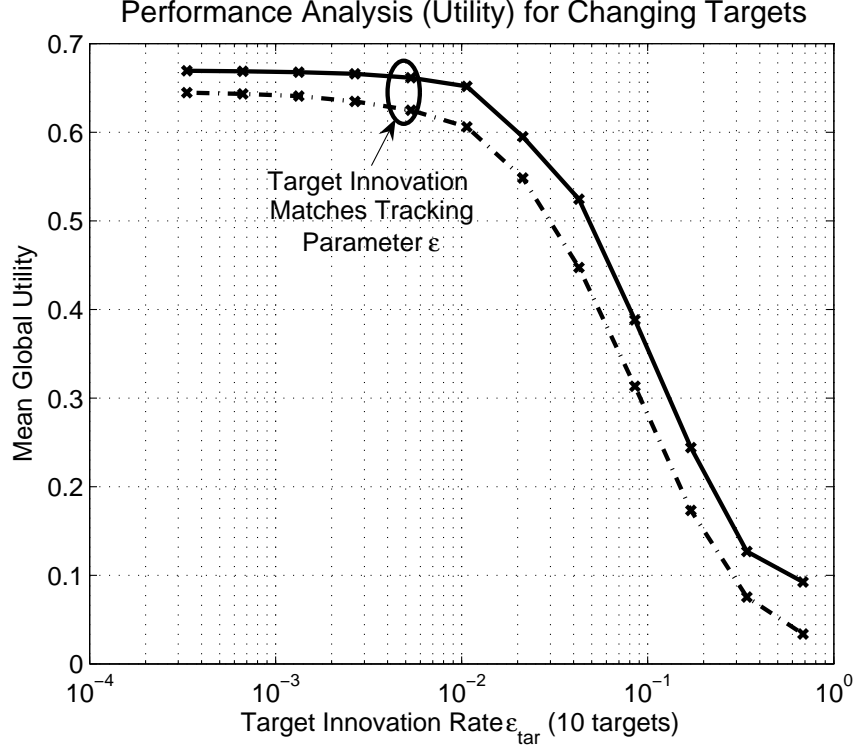


Figure 4.2: Utility is only high when target locations change sufficiently slowly. The solid line represents the case where the regret θ_n^l is reset as in Section 4.3.4. The broken line represents performance without resetting the regret. The target innovation rate is plotted on a logarithmic scale.

4.4.1 Tracking Dynamic Targets: Comparison of Performance

We first investigate the performance of Algorithm 4.2.2 when target vector \mathbf{r}_n evolves according to a slow process; at each iteration we allow each of the $M = 10$ targets to randomly jump to a new location with probability ρ . The overall utility (4.6) is shown in Figure 4.2 for various ρ , and the proportion of time spent by sensors in non-equilibrium states is shown in Figure 4.3. We also show the effect of the second part of Algorithm 4.3.1 (solid line), which enhances performance by periodically resetting the regret parameters.

As shown, utility decreases as targets become more mobile, since sensors require time to adapt to the constantly changing conditions, and hence may lose track of fast-moving targets. This is also shown by the time sensors spend out of equilibrium while adapting, which is a measure of the time taken to adapt to target changes. As predicted, the algorithm

Performance Analysis (Time in Equilibrium) for Changing Targets

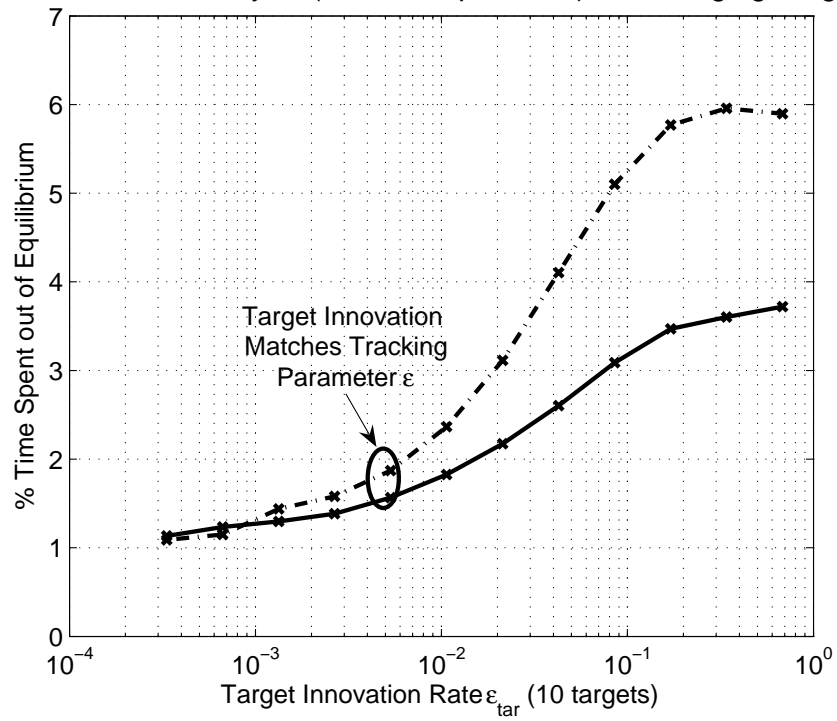


Figure 4.3: As target motion increases, the proportion of time spent in suboptimal states increases. The solid line represents the case where the regret θ_n^l is reset as in Section 4.3.4. The broken line represents performance without resetting the regret. The target innovation rate is plotted on a logarithmic scale.

Table 4.2: The Regret-based approach outperforms a simple greedy scheme. Performance is shown for optimal stepsize ε , and is defined as the average end-user utility per joule of energy spent in an iteration, per target.

Target Behaviour	Performance (Greedy)	Performance (Regret Tracking)
$N_{tar} = 1, \rho = 0.01$	82.9	96.4
$N_{tar} = 1, \rho = 0.05$	74.5	89.5
$N_{tar} = 10, \rho = 0.005$	107.2	112.6

performs well for slowly moving targets, but performance degrades for faster targets, since dynamics are not accounted for.

To show the benefit of the game theoretic approach, we also compare sensor activation according to Algorithm 4.2.2 with a simple greedy scheme, in which sensors activate deterministically if their average utility for activating in previous periods was positive. Table 4.2 compares performance in terms of the global utility per Joule of energy spent for three scenarios: First, with a single, slow-moving target with $\rho = 0.01$, then for a faster target with $\rho = 0.05$, and finally for ten targets, each with $\rho = 0.005$.

Table 4.2 shows that the regret-based scheme outperforms the simple greedy scheme in all cases. The key to this result is the randomization of actions in Algorithm 4.2.2, which allows sensors to properly coordinate actions, and is critical to the convergence properties the scheme.

4.4.2 Parameter Selection and Sensor Estimation

A key factor dictating sensor behaviour is the weight of the trade-off between a sensor's reward and the energy cost for activation. Parameter K_1 in (4.7) fulfills this role, and one may think of it as a price paid to a sensor per unit of information delivered. Naturally, increasing K_1 provides a greater incentive for sensors to activate, even when their signal is weak, thereby increasing both the amount of energy consumed and the information provided. However, since the user's utility is concave, there is a diminishing return for setting K_1 too high, which suggests a trade-off between performance and energy expended exists. We illustrate this in Figure 4.4, which shows the effect of varying the sensors' utility (pricing) parameter K_1 on both the end-user utility and energy consumed in the network.

Another factor influencing sensor behaviour is the accuracy of their awareness of nearby sensor activity. In particular, one may ask whether Algorithm 4.2.1, which delivers a maximum likelihood estimate of sensor activity, is effective. To address this, Figure 4.4 shows sensor network behaviour with the estimate provided by Algorithm 4.2.1, as well as the the-

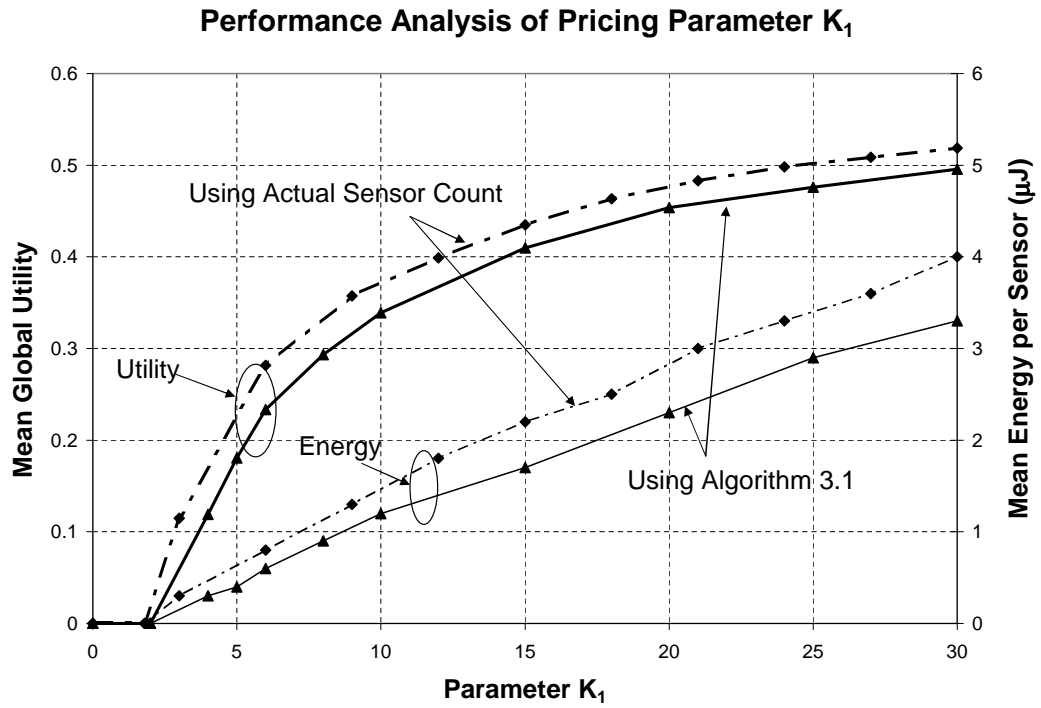


Figure 4.4: End user utility and energy increase as the incentive parameter for activating, K_1 , is increased. Overall performance is only degraded slightly when sensors estimate contention levels via Algorithm 4.2.1.

oretical optimal performance if the true number of active sensors were known. As shown, the degradation due to estimation is small; the general effect of using Algorithm 4.2.1 is that marginally fewer sensors are active on average (which actually reduces energy consumption).

The final innovation we illustrate is the timer-based load-sharing mechanism of Algorithm 4.3.1. This is intended to provide an additional layer of activation control to “smooth out” sensor activity so as to extend network lifetime, with minimal impact to performance.

Figure 4.5 shows both sensor network performance and energy consumption for a range of timeout parameters. Recall that when a sensor is active or inactive for a number of consecutive decision periods equal to the timeout parameter, it switches its pricing parameter K_1 to a smaller or larger value (respectively), effectively sending the sensor in and out of a backoff state. While large timeout parameters correspond to this feature being disabled, smaller timeout lengths cause the most well-situated sensors to back off after a period of time. This negatively impacts performance, since less well-positioned sensors are forced to operate, but the impact is limited, relative to the impact of parameter K_1 above. However, as shown, the mean energy spent and the energy standard deviation are also decreased for smaller timeout parameters. As before, this implies a trade-off between performance and energy usage which must be achieved by setting the timeout parameters appropriately.

4.5 Discussion

Energy efficient design for unattended ground sensor networks (UGSN) requires a highly integrated approach. Increasing the autonomy and intelligence of sensor nodes is critical to reducing costly communication overhead associated with coordinating sensor nodes, but this must be done carefully, as it is complicated for a large number of nodes to adapt simultaneously.

To address this, we have presented a computationally efficient adaptive filtering algorithm for effective node activation without costly overhead, and demonstrated the usefulness of tools used such as correlated equilibria of non-cooperative games and stochastic approximation. Some of the critical issues considered are a mechanism for sensor awareness, and the tracking capabilities of the regret-based algorithm for sensor deployment.

The stochastic approximation algorithms used here are extremely flexible, and can be applied to a large number of decentralized adaptive processing problems. Application of these techniques is an area for further study.

We conclude with a discussion of two useful extensions of the methods in this paper.

1. *Stochastic Sensor Scheduling*: Typically, a sensor node has a suite of sensors available

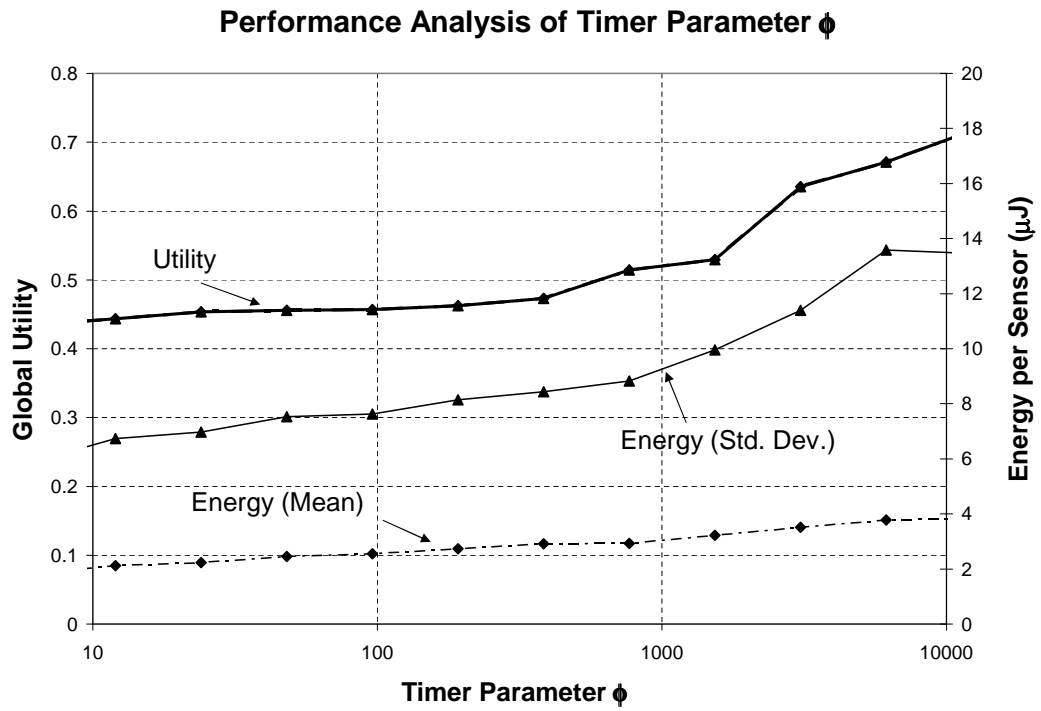


Figure 4.5: Utility is higher when sensors are not required to enter a backoff state, due to timeout. Energy consumption is reduced and more fairly distributed if sensors are forced to share duties due to the timeout/backoff mechanism.

for measurements, but can only deploy one or a few of these at a time. It is possible to extend our methods to the stochastic *sensor scheduling* problem, which deals with how these on-board sensors should be dynamically scheduled to optimize a utility function comprising of the Bayesian estimate of the target and the usage cost of the sensors. Such problems are partially observed stochastic control problems [137]. We refer the reader to [150] for the partially observed Markov decision process (POMDP) sensor scheduling problem.

2. *Analysis of stochastic approximation algorithms:* With constant stepsize, Algorithm 4.2.2 tracks a correlated equilibrium set that varies due to a slowly moving target. Suppose we were to assume that the target moves according to a slow Markov chain with transition probability matrix $I + \epsilon Q$ (where $\epsilon > 0$ is a small parameter and Q is a generator matrix with each row summing to zero). With this assumption, how can one analyze the tracking performance of Algorithm 4.2.2 with (the same) step size ϵ ? In our recent work [154], we have shown that the limiting behaviour (see [127]) of similar systems is captured by a Markovian switched ordinary differential equation, and we conjecture that a similar result applies to Algorithm 4.2.2, using a differential inclusion. This analysis requires use of yet another extremely powerful tool in stochastic analysis namely, the so called “martingale problem” of Strook and Varadhan, see [132] for comprehensive treatments of this area.

Bibliography

- [122] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks*, 38(4):393–422, 2002.
- [123] Martin Andersson, Dan Henriksson, Anton Cervin, and Karl-Erik rzn. Simulation of wireless networked control systems. *Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference*, 2005.
- [124] R. Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1:67–96, 1974.
- [125] R. Aumann. Correlated equilibrium as an expression of bayesian rationality. *Econometrica*, 55(1):1–18, 1987.
- [126] M. Benaïm, J. Hofbauer, and S. Sorin. Stochastic approximations and differential inclusions ii: Applications. Levine’s bibliography, UCLA Department of Economics, May 2005.
- [127] A. Benveniste, M. Metivier, and P. Priouret. *Adaptive Algorithms and Stochastic Approximations*, volume 22 of *Applications of Mathematics*. Springer-Verlag, 1990.
- [128] D. Bertsekas and J.N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA., 1996.
- [129] N. Bulusu, D. Estrin, L. Girod, and J. Heidemann. Scalable coordination for wireless sensor networks: Selfconfiguring localization systems. in *Proceedings of the Sixth International Symposium on Communication Theory and Applications (ISCTA 01)*, July 2001.
- [130] D. Fudenberg and D. Levine. *The theory of learning in games*. MIT Press, 1999.
- [131] A.J. Goldsmith and S.B. Wicker. Design challenges for energy-constrained ad hoc wireless networks. *IEEE Wireless Comm.*, 9(4):8–27, 2002.

- [132] Kushner H. *Approximation and Weak Convergence Methods for Random Processes, with applications to Stochastic Systems Theory*. MIT Press, Cambridge, MA., 1984.
- [133] S. Hart and A. Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.
- [134] S. Hart and A. Mas-Colell. *A reinforcement procedure leading to correlated equilibrium, in Economic Essays*, pages 181–200. Springer, 2001.
- [135] S. Hart and A. Mas-Colell. Uncoupled dynamics do not lead to nash equilibrium. *American Economic Review*, 93(5):1830–1836, December 2003.
- [136] J. Hill and D. Culler. Mica: A wireless platform for deeply embedded networks. *IEEE Micro*, 22(6):12–24, 2002.
- [137] Baras J and Bensoussan A. Optimal sensor scheduling in nonlinear filtering of diffusion processes. *SIAM Journal Control and Optimization*, 27(4):786–813, 1989.
- [138] R. Kannan, S. Sarangi, S. S. Iyengar, and Lydia Ray. Sensor-centric quality of routing in sensor networks. *IEEE Infocom*, 2003.
- [139] V. Krishnamurthy and G. Yin. Recursive algorithms for estimation of hidden Markov models and autoregressive models with Markov regime. 48(2):458–476, February 2002.
- [140] H.J. Kushner and G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer-Verlag, New York, NY, second edition, 2003.
- [141] Ning Li and Jennifer Hou. Localized fault-tolerant topology control in wireless ad hoc networks. *IEEE Transactions on Parallel and Distributed Computing*, 17(4):307–320, 2006.
- [142] A.B. MacKenzie and S.B. Wicker. Game theory and the design of self-configuring, adaptive wireless networks. *IEEE Communications Magazine*, pages 126–131, November 2001.
- [143] A.B. MacKenzie and S.B. Wicker. Stability of multipacket slotted aloha with selfish users and perfect information. *in Proceedings of IEEE INFOCOM*, 2003.
- [144] Heather Marandola, Joseph Mollo, and Paul Walter. Rembass-ii: the status and evolution of the army’s unattended ground sensor system. *Proceedings of SPIE*, 4743:99–107, 2002.

- [145] W.M. Merrill, F. Newberg, K. Sohrabi, W. Kaiser, and G. Pottie. Collaborative networking requirements for unattended ground sensor systems. *In Proc. IEEE Aerospace Conference*, 2003.
- [146] J. Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.
- [147] R. Nau, S. Gomez Canovas, and P. Hansen. On the geometry of nash equilibria and correlated equilibria. *International Journal of Game Theory*, 32(4):443–453, 2004.
- [148] J. Spall. *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. Wiley Press, 2003.
- [149] N. Srour. Unattended ground sensors: a prospective for operational needs and requirements. *Tech. rep., Army Research Laboratory*, October 1999.
- [150] Krishnamurthy V. Algorithms for optimal scheduling and management of hidden Markov model sensors. *IEEE Trans. Signal Proc.*, 50(6):1382–1397, 2002.
- [151] Yiping Xing and R. Chandarmouli. Distributed discrete power control for bursty transmissions over wireless data networks. In *in Proc. CISS?* 2004.
- [152] G. Yin and V. Krishnamurthy. Lms algorithms for tracking slow markov chains with applications to hidden markov estimation and adaptive multiuser detection. *IEEE Trans. Information Theory*, 51:2475–2490, 2005.
- [153] G. Yin and Q. Zhang. *Discrete-time Markov Chains: Two-time-scale Methods and Applications*. Springer, New York, NY, 2005.
- [154] Krishnamurthy V Yin G and Ion C. Regime switching stochastic approximation algorithms with application to adaptive discrete stochastic optimization. *SIAM Journal on Optimization*, 14(4):117–1215, 2004.

Chapter 5

Game Theoretic Dynamic Spectrum Access for Cognitive Radios ⁷

5.1 Introduction

Technologies such as mobile computing and cellular telephony are increasingly striving to deliver an “always connected” user experience. As these technologies become more ubiquitous, it becomes critical to make efficient use of limited radio resources to reliably deliver this experience to as wide a market as possible. This in turn requires active management of spectral resources, a challenge considering the decentralized structure of the radio system. To address this, researchers in cognitive radio [166, 168], propose RF devices that actively monitor and adjust to their radio environment to efficiently communicate in a crowded spectrum. This *dynamic spectrum access* functionality, in which cognitive radios compete for resources while respecting legacy (licensed) users, is the subject of this chapter.

We explore the spectrum overlay approach (also referred to as opportunistic spectrum access) to dynamic spectrum access, using a game theoretic framework to highlight issues of cooperation and competition between multiple radios. In this model, cognitive radios share portions of the RF spectrum (channels) that are temporarily unoccupied by licensed users. Each radio dynamically selects several available channels so as to balance its own demand (competition) against system-imposed sharing incentives (cooperation). Selections are made independently by each radio, based only on their own performance history. We focus on applications where primary users’ spectrum access activities either vary slowly with time (see [171, 172]), or where their spectrum access activities vary quickly, but average behaviour varies slowly. Example applications include the reuse of certain TV-bands that

⁷A version of this chapter has been submitted for publication. Maskery, M. and Krishnamurthy, V. Decentralized Dynamic Spectrum Access for Cognitive Radios: Cooperative Design of a Non-cooperative Game, IEEE Transactions on Communications.

are not used for TV broadcast in a particular region.

Since optimal resource allocation in a decentralized, competitive environment is not straightforward, we propose to operate radios according to a game theoretic algorithm which slowly adapts resource allocation over time. We show that this algorithm tracks the time-varying set of *correlated equilibrium* actions of the game, so that each radio learns to respond optimally to its environment. For appropriate radio utilities, this equilibrium leads to efficient global use of resources.

There are several reasons for using a game theoretic approach. First, since game theory explicitly recognizes the interdependence between radios, it can be used as a *synthesis tool* to provide decentralized algorithms for adaptive resource allocation. Second, the game theoretic concept of equilibrium provides a useful *analysis tool*; if we specify a simple algorithm that converges to equilibrium, then we can characterize the long-run behaviour of the system, which may be measured against a global, system-wide objective.

In this chapter, we assume that detecting the activities of primary users is sufficiently accurate that interference to primary users is below the required level. To resolve contention among cognitive radios, we use CSMA (carrier sense multiple access) to randomly allocate channel times among competing cognitive radios based on a reservation system. This simple mechanism allows us to capture fundamental issues such as uncertainty of the activity of others and nonlinear channel degradation due to crowding effects.

Related Work

Dynamic spectrum access presents technical challenges across the entire networking protocol stack. An overview of challenges and recent developments in dynamic spectrum access can be found in [176]. In the context of spectrum overlay, basic design components include spectrum opportunity identification and spectrum opportunity exploitation. The opportunity identification module is responsible for accurately identifying and intelligently tracking idle frequency bands that may be dynamic in both time and space. The opportunity exploitation module takes input from the opportunity identification module and decides whether and how a transmission should take place.

Spectrum opportunity detection can be reduced to a classic signal processing problem: detecting the presence of primary users' signals. Based on the cognitive radios' knowledge of the signal characteristics of primary users, three traditional signal detection techniques can be employed: matched filter, energy detector (radiometer), and cyclostationary feature detector [158]. While classic signal detection techniques exist in the literature, detecting primary transmitters in a dynamic wireless environment with noise uncertainty, shadowing,

and fading is a challenging problem that has attracted much research attention [159, 162, 174]. When the activities of primary users are fast varying, spectrum opportunity tracking becomes a critical issue. This problem is addressed within the framework of Partially Observable Markov Decision Processes (POMDP) in [177].

Once spectrum opportunities are detected, cognitive radios need to decide whether and how to exploit them. In the design of the spectrum exploitation module, specific issues include whether to transmit given that opportunity detectors may make mistakes, what modulation and transmission power to use, and how to share opportunities among secondary users to achieve a network-level objective.

The optimal design of spectrum access strategies in the presence of spectrum sensing errors has been addressed in [160, 161]. Specifically, the interaction between the spectrum access protocols at the MAC layer and the operating characteristics of the spectrum opportunity detector at the physical layer is quantitatively characterized, and the optimal joint design of opportunity detectors, access strategies, and opportunity tracking strategies is obtained. Orthogonal frequency division multiplexing (OFDM) has been considered as an attractive candidate for modulation in spectrum overlay networks as discussed in [157, 173]. Power control for cognitive radios needs to take into account the detection range of the opportunity detector, the maximum allowable interference level, and the transmission power of primary users [175]. This complex networking issue remains largely open.

Spectrum opportunity sharing among cognitive radios, which is the focus of this chapter, has been addressed in the literature. The problem of noncooperative radio resource allocation is considered in [171, 172] and related work from a non-game theoretic perspective, and in [169] from a game theoretic one, using a similar approach to the one presented here. However, [169] and similar references deal with a CDMA scenario which gives rise to a highly-structured supermodular game scenario, whereas the problem considered here lacks such convenient structure.

Organization of Chapter

There are several underlying themes in this chapter. Based on the model of Section 5.2, we investigate how radios can estimate information about their environment in Section 5.3. The issue of performance evaluation based on this information is dealt with in Section 5.4, which includes consideration of whether radios are designed for selfish or cooperative behaviour. Section 5.5 discusses our main decentralized adaptive algorithm for spectrum access, along with some related variants. These are compared through simulation in Section 5.6, and the main procedure is shown to be superior. Concluding remarks are made in Section 5.7.

5.2 Opportunistic Spectrum Access Model

We consider a network of fully connected cognitive radios communicating by exploiting channels unused by primary users. We divide time into equal slots of length Λ , and label slots $n = 1, 2, \dots$ (we also refer to a slot as a decision period). At the beginning of the n^{th} time slot, each cognitive radio $l = 1, 2, \dots, L$ knows the following:

1. \mathcal{C} , the number of channels in the radio system.
2. $C_n \in \mathbb{R}^{\mathcal{C}}$, the channel quality vector (bits per time slot for each channel) at time n .
3. $\mathbf{Y}_n \in \{0, 1\}^{\mathcal{C}}$, the current channel usage pattern of primary users; Channel i is in use if $\mathbf{Y}_n(i) = 1$.
4. $d_n^l \in \mathbb{R}$, the current demand level of cognitive radio l (in bits per time slot).
5. $m^l \in \mathbb{Z}^+$, the maximum number of channels that l may use simultaneously.

All these quantities are static or vary slowly in time. The time-dependence of C_n allows us to consider sharing channels with fast primary users, whose *statistics* are slowly varying. A fast primary user on channel i periodically preempts cognitive radio activity, reducing the effective quality to $C_n(i)$. We will refer to these specifically as fast primary users throughout the chapter, while the term “primary user” will be reserved for the slow varying type.

Next, each radio l chooses a channel allocation \mathbf{X}_n^l (see Section 5.5) from the state space:

$$\mathbf{S}_n^l = \left\{ x \in \{0, 1\}^{\mathcal{C}} : x \cdot \mathbf{Y}_n = 0, \sum x(i) \leq m^l \right\} \quad (5.1)$$

That is, each radio can select up to m^l any unused (by primary users) channels. We denote the joint action of all cognitive radios by $\mathbf{X}_n \in \mathbb{S}_n = \mathbf{S}_n^1 \times \dots \times \mathbf{S}_n^L$.

Cognitive radios share channels using a simple carrier sensing multiple access (CSMA) scheme, as follows: Divide each decision period n into K equal subslots, labeled n_1, n_2, \dots, n_K . (So each subslot has length Λ/K .) For every subslot n_k and channel i such that $\mathbf{X}_n^l(i) = 1$, radio l executes the following:

1. Generate a backoff time $\tau_{n_k}^l(i)$ according to a uniform distribution on the interval $(0, \tau_{\max})$ for some fixed parameter τ_{\max} .
2. Upon expiry of the backoff timer, monitor channel i and transmit data only if the channel is sensed clear.

Exactly one radio will transmit successfully on Channel i in subslot n_k , provided that its backoff time is sufficiently less than the next smallest time (allowing time to sense the channel clear and switch from receive to transmit mode). Otherwise there will be a collision on i for that subslot.

For each $n, i : \mathbf{X}_n^l(i) = 1$, and $k = 1, 2, \dots, K$, denote success in subslot n_k by:

$$\gamma_{n_k}^l(i) = I\{\text{channel } i \text{ captured by } l \text{ in subslot } n_k\}, \quad (5.2)$$

where $I\{\cdot\}$ is the usual indicator function. At the end of decision period n (of length Λ), each radio l will have collected the following information:

$$(\gamma_n^l(i), \tau_n^l(i)) = \left\{ (\gamma_{n_k}^l(i), \tau_{n_k}^l(i)) : \mathbf{X}_n^l(i) = 1, k = 1, 2, \dots, K \right\} \quad (5.3)$$

This information will be used for adaptive decision making in subsequent sections.

A block diagram of our proposed system is given in Figure 5.1. There are three time scales in our problem formulation. The slowest time scale corresponds to the variation of primary user activity and demand levels. Second, and much faster, is the decision time scale (intervals of length Λ) of the cognitive radios themselves, and third, the fastest time scale (intervals of length Λ/K), are the CSMA channel access attempts.

5.3 Estimating Channel Characteristics

It is clearly undesirable that cognitive radios compete for the same channel while other channels lie idle. It is thus crucial that a cognitive radio recognize and avoid crowded channels. To address this, we discuss how any radio user l may use the CSMA feedback (γ_n^l, τ_n^l) to estimate competition for channels, and calculate the expected throughput and number of collisions experienced by a radio, which will be used to adjust channel allocation decisions.

5.3.1 Channel Contention Estimate

To estimate $N_n^l(i)$, the number of users competing with l for channel i during decision slot n , consider a single CSMA channel access attempt on Channel i . Suppose there are $\phi(i)$ other users, each choosing a random backoff time $\tau^m(i)$ uniformly on $(0, \tau_{\max})$. If l chooses $\tau^l(i) = t$, it captures the channel if $t < \tau^m(i) + \delta$ for all $m \neq l$, where δ is the time required to sense the channel clear and switch its receiver from receive to transmit mode. Let $\tau_{\phi(i)}^{(1)}$

denote the smallest of the $\phi(i)$ backoff times (the first order statistic). It is well known that:

$$Pr(l \text{ captures channel}) = Pr\left(\tau_{\phi(i)}^{(1)} > t + \delta\right) = \begin{cases} \left(1 - \frac{t+\delta}{\tau_{\max}}\right)^{\phi(i)}, & t \leq \tau_{\max} - \delta. \\ 0, & t > \tau_{\max} - \delta. \end{cases} \quad (5.4)$$

Since $\phi(i)$ is fixed, and CSMA attempts are independent between subslots, radio l can compute the likelihood of contention level $\phi(i)$ as:

$$\begin{aligned} L(\phi(i)) &= \prod_{k:\gamma_{n_k}^l(i)=1} Pr\left(\tau_{\phi(i)}^{(1)} > \tau_{n_k}^l(i) + \delta\right) \cdot \prod_{k:\gamma_{n_k}^l(i)=0} Pr\left(\tau_{\phi(i)}^{(1)} < \tau_{n_k}^l(i) + \delta\right), \\ &= \prod_{k:\gamma_{n_k}^l(i)=1} \left(1 - \frac{\tau_{n_k}^l(i) + \delta}{\tau_{\max}}\right)^{\phi(i)} \cdot \prod_{k:\gamma_{n_k}^l(i)=0} \left(1 - \left(1 - \frac{\tau_{n_k}^l(i) + \delta}{\tau_{\max}}\right)^{\phi(i)}\right). \end{aligned} \quad (5.5)$$

The maximum likelihood estimate $\hat{N}_n^l(i)$ of the number of competing users obtained by maximizing (5.5), i.e. solving:

$$\sum_{k:\gamma_{n_k}^l(i)=0} \frac{a_k^{\phi(i)} \log(a_k)}{1 - a_k^{\phi(i)}} = \sum_{k:\gamma_{n_k}^l(i)=1} \log(a_k), \quad (5.6)$$

where $a_k = 1 - (\tau_k^l(i) + \delta)/\tau_{\max}$.

Eq. (5.6) is difficult to solve, but we characterize the average behaviour numerically in Figure 5.2, which shows that $\hat{N}_n^l(i)$ increases with the number of channel access failures, decreases when the maximum successful backoff time among the K subslots increases, and increases when the minimum unsuccessful backoff time among the K subslots decreases.

By replacing the a_k terms on the left-hand side of (5.6) by their average, it is possible to approximate $\hat{N}_n^l(i)$ by:

$$\hat{N}_n^l(i) \approx -\log\left(1 + \frac{|I_0| \log(\bar{a}_0)}{\sum_{k:\gamma_{n_k}^l(i)=1} \log(a_k)}\right) / \log(\bar{a}_0), \quad (5.7)$$

where $|I_0| = \sum_{k=1}^K (1 - \gamma_{n_k}^l(i))$, and $\bar{a}_0 = (\sum_{k:\gamma_{n_k}^l(i)=0} a_k) / |I_0|$. Numerical studies show that (5.7) is quite accurate on average, but may have large error when either $|I_0|$ or the successful backoff times are large. In this case, we recommend using (5.7) to generate an initial guess, which may be refined by the Newton-Raphson method.

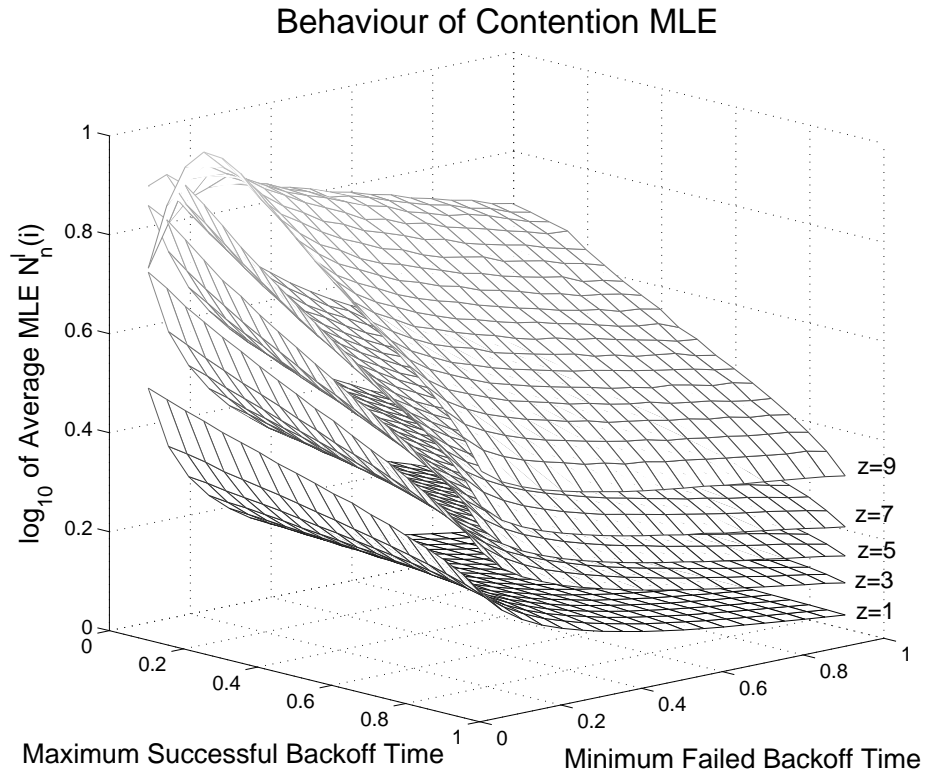


Figure 5.2: Average result of the contention estimate $\hat{N}_n^l(i)$. z denotes the number of failed CSMA attempts out of $K = 10$, and the maximum backoff time is $\tau_{\max} = 1$. Each data point represents an average of 5000 randomly generated observations with the specified maximum successful backoff time and minimum failed backoff time.

5.3.2 Other Channel Characteristics

Using contention estimates $\hat{N}_n^l(i)$, we can estimate the throughput and number of collisions on a channel (which are more relevant measures of a radio's performance), as follows:

Using (5.4) and the uniform distribution of backoff times, we may compute the unconditional (with respect to backoff time) probability of channel capture in a subslot as:

$$\begin{aligned} R_n^l(i) &= x^l(i) \int_0^{\tau_{\max}-\delta} \left(1 - \frac{t+\delta}{\tau_{\max}}\right)^{\hat{N}_n^l(i)} \frac{1}{\tau_{\max}} dt \\ &= \frac{x^l(i)}{1 + \hat{N}_n^l(i)} \left(1 - \frac{\delta}{\tau_{\max}}\right)^{1 + \hat{N}_n^l(i)}. \end{aligned} \quad (5.8)$$

(This formula holds for $\hat{N}_n^l(i) > 0$, otherwise $R_n^l(i) = 1$.)

Next, observe that l is involved in a channel collision in a subslot if either (a.) it has the lowest backoff time, but by a margin less than δ , or (b.) it does not have the lowest backoff time, but is within δ of the lowest. It follows that, for $\hat{N}_n^l(i) > 0$, the probability of l being involved in a collision on Channel i in subslot n_k , using backoff time $\tau_{n_k}^l = t$ is given by:

$$\begin{aligned} Q_n^l(i, t) &= Pr(t - \delta < \tau_N^{(1)} < t \text{ or } t < \tau_N^{(1)} < t + \delta) \\ &= Pr(\tau_N^{(1)} < t + \delta) - Pr(\tau_N^{(1)} < t - \delta) \\ &= x^l(i) \left[\left(1 - \frac{\max\{t - \delta, 0\}}{\tau_{\max}}\right)^{\hat{N}_n^l(i)} - \left(1 - \frac{\min\{t + \delta, \tau_{\max}\}}{\tau_{\max}}\right)^{\hat{N}_n^l(i)} \right]. \end{aligned} \quad (5.9)$$

Again, integrating out the backoff time gives the unconditional probability of collision:

$$Q_n^l(i) = \frac{x^l(i)\delta}{\tau_{\max}} + \frac{x^l(i)}{1 + \hat{N}_n^l(i)} \left[1 - \left(\frac{\delta}{\tau_{\max}}\right)^{1 + \hat{N}_n^l(i)} - \left(1 - \frac{\delta}{\tau_{\max}}\right)^{1 + \hat{N}_n^l(i)} \right]. \quad (5.10)$$

Since channel access attempts are i.i.d., $R_n^l(i)$ is also the expected proportion of successful CSMA attempts in a given decision period n , and $Q_n^l(i)$ is the expected proportion of CSMA attempts that result in collisions during period n . These quantities will be very useful for measuring radio performance in the next section.

5.4 System Performance and Radio Utility

The goal of this chapter is to achieve a *global* objective (efficient allocation of radio resources) using a *decentralized* scheme (local adaptation by individual radios). Consequently, we must demonstrate a connection between a global utility and the local utility function that will guide the allocation decisions of each radio in Section 5.5. This connection is presented below through the derivation of global (Section 5.4.1) and local (Section 5.4.2) performance measures.

5.4.1 Global System Utility

When each cognitive radio has roughly the same priority and bandwidth requirements, a reasonable global objective is for channel resources to be allocated fairly, such that the proportion of resources captured by the worst-off cognitive radio (relative to its demand) is maximized (max-min fairness). That is, the global system utility in decision period n is:

$$U(x) = \min_{l=1,\dots,L} \left(\min \left(\frac{C_n^T R_n^l}{d^l}, 1 \right) \right), \quad (5.11)$$

where each term $C_n^T R_n^l / d^l$ represents the “satisfaction level” of Radio m . This is the total amount of resource captured ($C_n^T R_n^l$) divided by the demand level d^l .

Substituting (5.8) into (5.11) yields an alternative formulation for the global utility:

$$U(x) = \min_{l=1,\dots,L} \left(\min \left(\frac{1}{d^l} \sum_i C_n(i) \frac{x^l(i)}{N_n^l(i) + 1} \left(1 - \frac{\delta}{\tau_{\max}} \right)^{1+N_n^l(i)}, 1 \right) \right), \quad (5.12)$$

for given CSMA parameters (δ, τ_{\max}) .

The system objective is to maximize $U(x)$ over all $x \in \mathbb{S}$, thus maximizing the satisfaction level of the worst-off user. However, for a decentralized implementation, we cannot choose an action from \mathbb{S} in a coordinated fashion, but each radio m must choose its own action $x^l \in \mathbf{S}^l$. Moreover, as we will see in the next section, the global utility cannot be easily evaluated by any one user, so an appropriate substitute must be found.

5.4.2 Local Radio Utility

If each cognitive radio had a reliable estimate of $U(x)$, then decentralized operation would be straightforward; each radio would simply act to maximize the global utility directly. Unfortunately, this would require l knowing d^m and x^m for all $m \neq l$, which is unrealistic.

We therefore construct an alternative utility function $u^l(x^l)$ which has the property that, if each radio l obtains a high value of $u^l(x^l)$, then the global utility $U(x)$ will also be high.

The first portion of l 's local utility reflects the self-interested component of (5.12):

$$u^l[0](x^l) = \min \left(\frac{1}{d^l} \sum_i C_n(i) \frac{x^l(i)}{x^l(i) + \hat{N}_n^l(i)} \left(1 - \frac{\delta}{\tau_{\max}} \right)^{x^l(i) + \hat{N}_n^l(i)}, 1 \right). \quad (5.13)$$

Maximizing (5.13) directly maximizes l 's part of the global utility $U(x)$ (Note we have used $N^m(i) = N^l(i)$ if $x^m(i) = x^l(i)$ and $N^m(i) = N^l(i) - 1$ if $x^m(i) = 1, x^l(i) = 0$.)

A game with (5.13) as the only component of the utility function would resemble a classic congestion game, which might be readily solved in closed form. However, (5.13) neglects a good portion of the global objective; l should maximize others' satisfaction of demand as well as its own. Since we assume that d^m and x^m are unknown for $m \neq l$, we induce such cooperation through the following two principles:

- Radio l 's realized rate should not exceeds its demand d^l , as this leaves fewer resources for other users.
- Radio l should minimize the number of CSMA collisions it causes, as this impacts the performance of other users.

These principles can be justified by noting that the components of (5.12) belonging to $m \neq l$ are decreasing in $x^l(i)$.

To satisfy the first principle, we introduce a penalty for achieving excess rate:

$$u^l[1](x^l) = -\frac{1}{d^l} \left(C_n^T R_n^l - (d^l + \beta) \right)^+, \quad (5.14)$$

where $(y)^+$ denotes the operation $\max\{y, 0\}$, and parameter β represents the size of a "grace" region, where excess rate is not penalized. (Since l 's realized rate is observed in noise, it may "accidentally" satisfy more than its demand, so small excesses are not penalized.)

The second principle, to minimize collisions, is satisfied by considering $Q_n^l(i)$ in (5.10). Neglecting collisions involving three or more users, $Q_n^l(i)$ can be interpreted as the degradation of Channel i (proportional to $C_n(i)$) caused by l . If this is spread evenly among all users, we have that the average performance degradation seen by radio m , caused by activity of Radio l on Channel i is $D_n^l(i) = Q_n^l(i)/\hat{N}^l(i)$. Our penalty is a weighted sum of

the channel degradation:

$$u^l[2](x^l) = -\frac{1}{\sum_k C_n(k)} \sum_{i: \hat{N}^l(i) > 0} C_n(i) D_n^l(i). \quad (5.15)$$

The final utility function for Cognitive radio l is given by:

$$u^l(x^l) = \max \left\{ u^l[0](x^l) + \alpha_1 u^l[1](x^l) + \alpha_2 u^l[2](x^l), 0 \right\}, \quad (5.16)$$

Eq. (5.16), with user-defined parameters (α_1, α_2) , is used to guide channel allocations by allowing each radio to take action X_n^l , observe utility u_n^l , and generate a new action that increases its expected utility. The method for choosing actions is the subject of Section 5.5.

5.5 Decentralized Adaptive Channel Allocation

In this section we describe our decentralized learning approach to opportunistic spectrum access. Our approach is based on the regret matching procedure of [163], formulated as a distributed stochastic approximation algorithm. This allows us to specify an adaptive variant of the original procedure, called “Regret Tracking,” which uses a constant stepsize to dynamically adapt to time varying conditions. The procedure is game theoretic in nature; it converges even when multiple cognitive radio users are simultaneously adapting their behaviour. This is critical observation; since naive, single-agent procedures do not account for the presence of multiple users, and hence may not converge.

In the regret tracking procedure, each radio takes a sequence of actions $\{\mathbf{X}_n^l \in \mathbf{S}_n^l : n = 0, 1, 2, \dots\}$ and observes a sequence of rewards $\{u_n^l \in \mathbb{R} : n = 0, 1, 2, \dots\}$. The action at time $n + 1$ is a random function of this history of actions and rewards.

Before proceeding, we clarify a point of notation. In (5.16), a radio’s utility u^l is written only as a function of its own action. However, it is also implicitly a function of the channel contention $\hat{N}^l(i)$, which depends on the actions of other players. In game theory, this is made explicit by writing $u^l(\mathbf{X}_n^l, \mathbf{X}_n^{-l})$, and we follow this convention below.

Recall the channel allocation of radio l at decision period n is denoted $\mathbf{X}_n^l \in \mathbf{S}^l$, with the joint allocation of all radios denoted $\mathbf{X}_n \in \mathbb{S}$. Based on the history of allocations and utilities, radio l compute average regret values $\theta_{n,jk}^l : j, k \in \mathbf{S}^l$, according to:

$$\theta_{n,jk}^l = \sum_{\tau \leq n: \mathbf{X}_\tau^l = j} \varepsilon_{\tau-1} \left(\prod_{\sigma=\tau}^{n-1} (1 - \varepsilon_\sigma) \right) \left(u^l(k, \mathbf{X}_\tau^{-l}) - u^l(j, \mathbf{X}_\tau^{-l}) \right). \quad (5.17)$$

Depending on whether $\varepsilon_\tau = 1/(\tau + 1)$ or $\varepsilon_\tau = \varepsilon$, this is a simple or moving average:

$$\begin{aligned}\theta_{n,jk}^l &= \frac{1}{n} \sum_{\tau \leq n: \mathbf{X}_\tau^l = j} \left(u^l(k, \mathbf{X}_\tau^{-l}) - u^l(j, \mathbf{X}_\tau^{-l}) \right), \\ \theta_{n,jk}^l &= \sum_{\tau \leq n: \mathbf{X}_\tau^l = j} \varepsilon(1 - \varepsilon)^{n-\tau} \left(u^l(k, \mathbf{X}_\tau^{-l}) - u^l(j, \mathbf{X}_\tau^{-l}) \right).\end{aligned}\quad (5.18)$$

The regret values measure the average gain that l would have received had he chosen allocation k in the past instead of j . If the gain is positive, then l is more likely to switch to k in the future. Note, however, that (5.17) requires that l knows what utility he would have received for each action, *even if that action was not taken*. We address this difficulty either by assuming that extra resources are available to monitor unused channels to reconstruct potential utilities, or by using the modified regret tracking algorithm of Section 5.5.2.

The exact channel allocation algorithm is summarized in Algorithm 5.5.1. This is a regret tracking procedure, executed independently by each radio.

Algorithm 5.5.1 Adaptive Learning for Channel Allocation: Define parameters $(u^l, \mu, \{\varepsilon_n : n = 1, 2, \dots\}, \theta_0^l, \mathbf{X}_0^l)$, where u^l are the radio utilities, μ satisfies

$$\mu > (S^l - 1)(u_{\max}^l - u_{\min}^l), l = 1, 2, \dots, L, \quad (5.19)$$

$((u_{\max}^l, u_{\min}^l)$ are obtained from (5.16)), $\{\varepsilon_n\}$ are small stepsizes, and $\theta_0^l, \mathbf{X}_0^l$ are arbitrary initial regrets and actions. Also define the $\mathbf{S}^l \times \mathbf{S}^l$ instantaneous regret matrix with entries:

$$\mathbf{H}_{jk}^l(\mathbf{X}_n) = I\{\mathbf{X}_n^l = j\} \left(u^l(k, \mathbf{X}_n^{-l}) - u^l(j, \mathbf{X}_n^{-l}) \right). \quad (5.20)$$

Each radio executes the following steps:

1. **Initialization:** Set $n = 0$, take action \mathbf{X}_0^l , and initialize $\theta_0^l = \mathbf{H}^l(\mathbf{X}_0)$. Set $n = 1$, take action $\mathbf{X}_1^l = \arg \max_k H_{jk}^l(\mathbf{X}_0)$, where $j = \mathbf{X}_0^l$, and set $\theta_1^l = \mathbf{H}^l(\mathbf{X}_1)$.
2. Repeat for $n = 2, 3, \dots$:
Action Update: Choose $\mathbf{X}_{n+1}^l = k$ with probability

$$Pr(\mathbf{X}_{n+1}^l = k | \mathbf{X}_n^l = j, \theta_n^l = \theta^l) = \begin{cases} \max\{\theta_{jk}^l, 0\}/\mu, & k \neq j, \\ 1 - \sum_{i \neq j} \max\{\theta_{ji}^l, 0\}/\mu, & k = j. \end{cases} \quad (5.21)$$

Average Regret Update: Given $\mathbf{H}^l(\mathbf{X}_{n+1})$, update θ_{n+1}^l according to the stochastic

approximation (SA):

$$\theta_{n+1}^l = \theta_n^l + \varepsilon_n(\mathbf{H}^l(\mathbf{X}_{n+1}) - \theta_n^l), \quad (5.22)$$

5.5.1 Convergence of Adaptive Channel Allocation

Because Algorithm 5.5.1 tends to switch to actions with higher rewards at each decision period, channel allocations converge such that no allocation with positive regret is played. Since the set of correlated equilibrium strategies [155] is, by definition, equivalent to the set of strategies with this no regret condition, it follows that allocation decisions converge to the set of correlated equilibrium under Algorithm 5.5.1.

For a decreasing stepsize $\varepsilon_n = 1/(n+1)$, almost sure convergence of Algorithm 5.5.1 to the set of correlated equilibria is proven in [156, 163] as $n \rightarrow \infty$. For a constant stepsize ε , this is replaced by weak convergence (i.e. convergence in distribution), as $\varepsilon \rightarrow 0$. This weak convergence result is a general property for constant stepsize stochastic approximation (SA) algorithms when decreasing stepsize SA convergence holds [167]. Intuitively then, our adaptive version of Algorithm 5.5.1 should also track the set of correlated equilibria, with changes to the utility functions handled smoothly by the constant stepsize.

To formally state the convergence result, we introduce the empirical distribution of play, which can be viewed as a diagnostic that monitors the performance of the entire cognitive radio network, as follows:

Definition 5.5.1 Let $e_x = [0, 0, \dots, 1, 0, \dots, 0]$ with a one in the x^{th} position. The empirical distribution of play to time n is:

$$\bar{z}_n = \sum_{\tau \leq n} \varepsilon_{\tau-1} \left(\prod_{\sigma=\tau}^{n-1} (1 - \varepsilon_{\sigma}) \right) e_{X_{\tau}}, \quad (5.23)$$

Here ε_n is the weighting factor given in Algorithm 5.5.1. Similarly to (5.18), \bar{z}_n can be viewed as an average or moving average frequency of play, with $\sum_i \bar{z}_n(i) = 1$. Moreover, as was the case for θ_n , \bar{z}_n may be represented by the recursion:

$$\bar{z}_{n+1} = \bar{z}_n + \varepsilon_n(e_{X_{n+1}} - \bar{z}_n), \quad (5.24)$$

where X_{n+1} given by Step (2a) of Algorithm 5.5.1.

We also need the notion of convergence to the *set* of correlated equilibria:

Definition 5.5.2 \bar{z}_n converges to the set of correlated equilibria, \mathcal{C}_e if, for any $\varepsilon > 0$, there

exists $N_0(\varepsilon)$ such that for all $n > N_0$, we can find $\psi \in \mathcal{C}_e$ at a distance less than ε from \bar{z}_n .

The convergence proof follows a differential inclusion approach, as in [156], suitably modified for a constant stepsize through the methods of [167].

The differential inclusion approach proceeds by introducing the following continuous time interpolated process for any $\varepsilon > 0$:

$$\bar{z}^\varepsilon(t) = \bar{z}_n \text{ for } t \in [\varepsilon n, \varepsilon(n+1)), \quad (5.25)$$

for $n = 0, 1, 2, \dots$. Next, define the following differential inclusion, as in [156]:

$$\frac{dz}{dt} \in \nu(z) \times \Delta S^{-l} - z, \quad (5.26)$$

where $\nu(z)$ is the set of probability distributions over \mathbf{S}^l , such that:

$$\sum_k \nu_k[\theta_{kj}^l(z)]^+ = \nu_j \sum_k [\theta_{jk}^l(z)]^+$$

for all $j, k \in \mathbf{S}^l$. Here ΔS^{-l} is the set of all probability distributions over joint actions of the $L - 1$ competing radios, and $\theta_{jk}^l(z)$ is the average regret corresponding to play history z .

The main result is stated as follows:

Theorem 5.5.1 *Consider the interpolated process (5.25) generated by Algorithm 5.5.1. If all radios operate according to this algorithm, the following results hold:*

1. *All solutions $z(t)$ to (5.26) converge to the set of correlated equilibria as $t \rightarrow \infty$.*
2. *For a decreasing stepsize $\varepsilon_n = 1/(n+1)$, the trajectory of the interpolated process $z^\varepsilon(t)$ converges almost surely to a trajectory $z(t)$ satisfying (5.26).*
3. *Under a constant stepsize $\varepsilon_n = \varepsilon$ in Algorithm 5.5.1, the trajectory of the interpolated process $z^\varepsilon(t)$ converges weakly as $\varepsilon \rightarrow 0$ to a trajectory $z(t)$ satisfying (5.26).*
4. *As $t \rightarrow \infty$, since $z(t)$ converges to the set of correlated equilibria, the trajectory $z^\varepsilon(t)$ also converges to this set.*

The proof of Theorem 5.5.1, parts (1), (2) and (4) is given in [156]. Part (3) follows from standard arguments as in [167].

5.5.2 Other Adaptive Strategies

For reference, we briefly review three alternative methods for decentralized decision making that can be applied to the opportunistic spectrum access problem; best response, fictitious play and modified regret tracking. We relate these to our regret tracking approach, and in future sections compare the three approaches through simulation.

Best Response In the simplest approach, each cognitive radio chooses the channel allocation that maximizes its utility, assuming that the actions of other cognitive radios will not change. In Algorithm 5.5.1, this corresponds to setting $\varepsilon_n = 1$ and replacing Step (2a) with $\mathbf{X}_{n+1}^l = \arg \max_k (H_{jk}^l(\mathbf{X}_n))$ where $j = \mathbf{X}_n^l$. In best response, the average history θ_n^l does not need to be tracked at all, since actions are based solely on feedback from the previous step. Nevertheless, it still provides a useful performance measure for the system since it indicates how close the system is to equilibrium.

Fictitious Play Best response suffers from the defect that actions of other users are assumed to be constant between iterations. This was realized very early on, and [170] proposed that users should respond not just to the last observation, but to the average history. Fictitious play has since been widely studied and has been shown to converge in many, but not all, games. In [165] it was shown that fictitious play is a special case of regret-based algorithms. Specifically, (assuming a decreasing stepsize ε_n), it corresponds to replacing Step (2a) in Algorithm 5.5.1 with $\mathbf{X}_{n+1}^l = \arg \max_k (\theta_{n,jk}^l(\mathbf{X}_n))$ where $j = \mathbf{X}_n^l$. An adaptive version of fictitious play, with constant stepsize ε , can also be generated in this manner.

Modified Regret Tracking The preceding strategies rely on each cognitive radio knowing the value of *all* actions, not just those actually taken. This means radios must monitor all possible channels in order to determine $\hat{N}_n^l(i)$ for all i and hence $u^l(k, \cdot)$ for all possible k . This can be achieved straightforwardly if extra receivers are used to scan channels not currently in use by the cognitive radio. An alternative procedure is proposed in [164], which replaces the explicit utility of actions not taken with an estimate, proceeds as follows. Label the probability distribution used to choose action \mathbf{X}_n^l in Step (2) of Algorithm 5.5.1 by p_n^l . We first replace (5.20) by:

$$\mathbf{H}_{jk}^l(\mathbf{X}_n) = I\{\mathbf{X}_n^l = k\} \frac{p_n^l(j)}{p_n^l(k)} u^l(k, \mathbf{X}_n^{-l}) - I\{\mathbf{X}_n^l = j\} u^l(j, \mathbf{X}_n^{-l}). \quad (5.27)$$

This avoids the need to evaluate u^l for actions not taken. However, we must compensate for this by allowing radios to explore alternative actions. We therefore replace Step (2) of Algorithm 5.5.1 by:

- 2') Generate a uniform random variable U . If $U < \delta$ (small), choose \mathbf{X}_{n+1}^l from a uniform distribution over \mathbf{S}^l . Otherwise, choose $\mathbf{X}_{n+1}^l = k$ with probability as in (5.21).

It is proven in [164] that the non-tracking version of this algorithm converges almost surely to the set of correlated equilibria. However, in our tracking simulations, convergence is much slower, due to the required periodic random exploration.

5.6 Numerical and Simulation Results

We now provide a numerical (Matlab) comparison of the methods of Section 5.5. For demonstration purposes, we specify a relatively small number of users (six) and channels (ten), and focus on moderately congested systems, with total capacity of channels roughly equal to total user demand. Similar results hold for other cases. We take $K = 20$ CSMA attempts per decision period on selected channels, and assume unused channels are scanned randomly such that information is obtained as if $K \approx 10$ CSMA attempts were made.

5.6.1 Effect of Utility Pricing on Performance of Regret Tracking

In this section we investigate the impact of (α_1, α_2) on system performance. (Recall (α_1, α_2) , parameterize the radio utility $u^l(x^l)$, and affect l 's level of cooperation with system objectives.) The problem of selecting optimal parameters can be viewed as a pricing problem (in this case carried out offline).

For each parameter choice (α_1, α_2) , 20 scenarios were generated with channel quality C_n and user demands d_n^l , selected from uniform distributions on $\{1, 2, 3\}$ and $\{1, 2, 3, 4\}$, respectively. Hence, on average there was 16 units of channel capacity and 18 units of demand, indicating mild system congestion. We held $m^l = 2$ for each radio, to limit the size of the decision space. For ideal operation, we initially assume that primary user activity is fixed to two specific channels. We will address varying channel occupancy in Section 5.6.3.

For each choice (α_1, α_2) , system performance (5.11) was averaged over 1000 iterations of Algorithm 5.5.1 and the 20 randomly selected scenarios. The result is shown in Figure 5.3. System performance was highest for $\alpha_1 \approx 0.3$ and $\alpha_2 > 1$. Hence completely selfish behaviour ($\alpha_1 = \alpha_2 = 0$) is not optimal. To simulate good cooperation, we chose $(\alpha_1, \alpha_2) = (0.2, 1.8)$ for subsequent analysis.

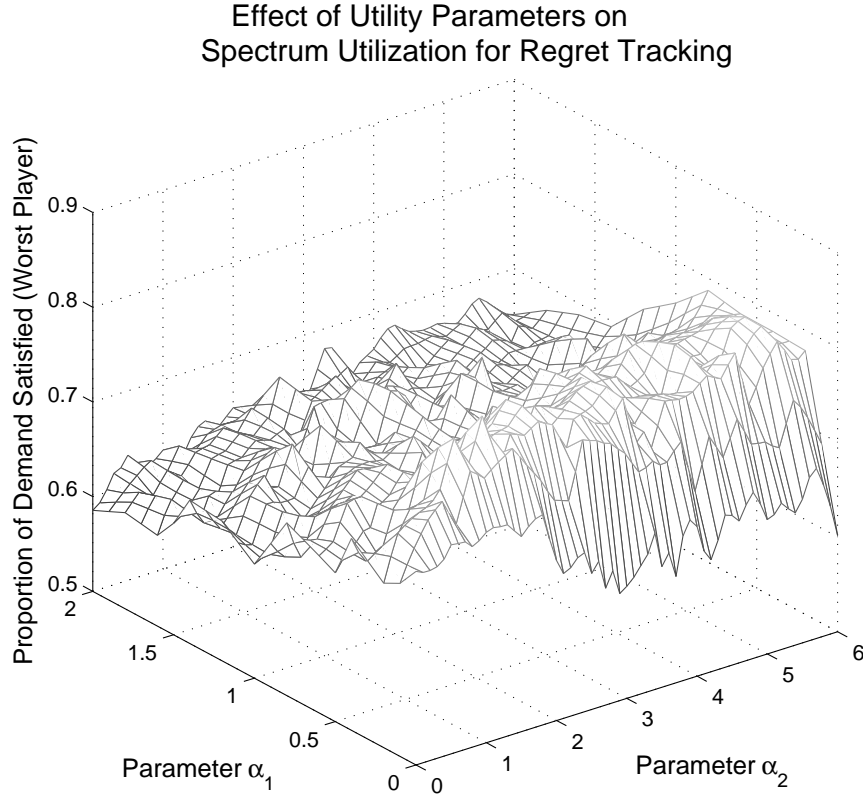


Figure 5.3: The average system performance for a fixed scenario type (mild congestion) depends on the parameter choices (α_1, α_2) of the radio utility (5.16). The parameters can be thought of as unit prices, imposed by a system-wide authority to discourage different types of interference. The selfish (anarchy) case, $\alpha_1 = \alpha_2 = 0$, does not yield the best system performance, but left to their own devices radios may gravitate towards this case.

It is interesting to ask whether a player can benefit by deviating from these cooperative parameters (set by design or by a system authority). The answer appears to be “yes”; if one player takes $(\alpha_1, \alpha_2) = (0, 0)$, while the others use $(\alpha_1, \alpha_2) = (0.2, 1.8)$, the selfish player achieves much higher satisfaction of demand (96.5% after 3000 iterations), at the expense of the system (the worst off player satisfies 76% of his demand). This points to the ultimate benefit of cooperative design; even as radios seek to maximize their own utility, this should reflect system, not individual, performance. It also suggests that the parameter-tuning “meta-game” is of the prisoner’s dilemma type, with Nash equilibrium $(\alpha_1, \alpha_2) = (0, 0)$.

5.6.2 Comparison of Algorithms in a Static Environment

We now compare the long-run performance of Algorithm 5.5.1 to other possible approaches assuming fixed cognitive radio demand d_n^l primary user activity Y_n .

Simulation of Algorithm 5.5.1 and the three alternatives in Section 5.5.2 were performed as follows. 100 scenarios were randomly generated as in Section 5.6.1. Each cognitive radio ran the same algorithm over 3000 iterations, (CSMA access, channel contention estimation, and adaptive channel allocation). The results are shown in Figures 5.4 and 5.5.

As shown, the regret tracking algorithm achieves the best performance, and most closely approaches the set of correlated equilibria of the spectrum access game (has the lowest regret). High utility was designed to correspond with good system performance, so it is not surprising that locally optimal (i.e. equilibrium) performs well.

Modified regret tracking performed significantly poorer, since radio awareness is severely impaired in this procedure (radios do not know the contention on a channel until they use it). Analysis reveals that the random exploration required by the modified procedure is chiefly to blame for its poor performance. (Exploration is done indifferently between good and bad actions, and may drastically impact the performance of other radios.)

The two remaining procedures, fictitious play and best response, performed surprisingly poorly, not even approaching 50% spectrum utilization. The problem here can be traced to the fact that fictitious play and regret tracking choose the *best* action, whereas regret tracking chooses randomly from among all *better* actions. This greedy behaviour performs poorly under noisy utility measurements, as is the case here, since users tend to overreact to bad information. This hypothesis was validated by running simulations with perfect utility information. In this case, all algorithms performed equally well and reached equilibrium almost immediately.

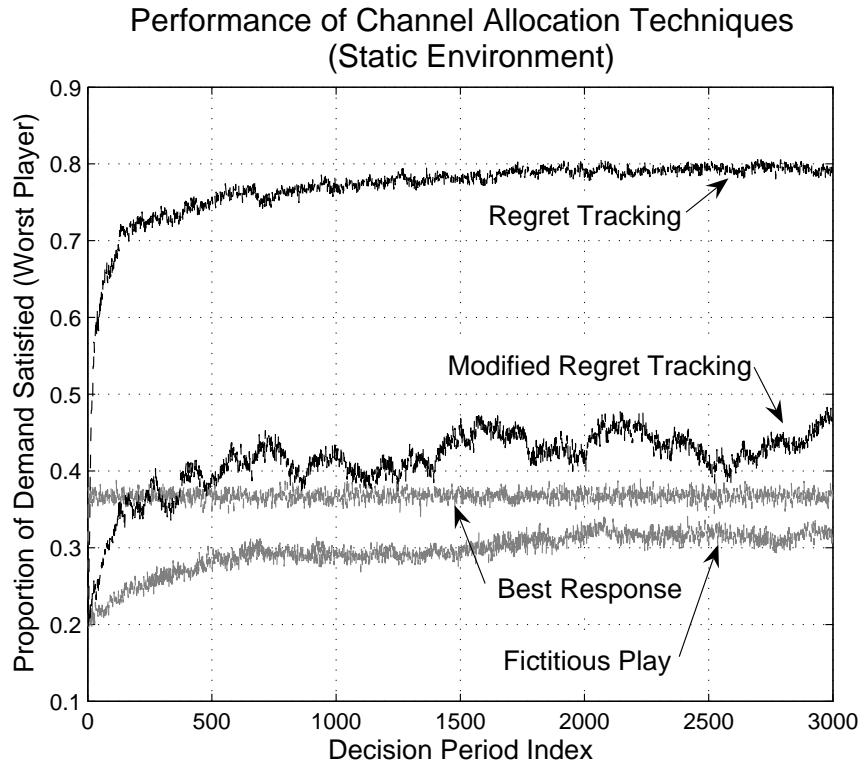


Figure 5.4: Performance comparison of the channel allocation techniques of Section 5.5, according to global design objective (5.11). The regret-based algorithms outperform the classical “greedy” algorithms due to their tolerance to noisy feedback. The optimal performance based on the selected scenarios is approximately 92.5%.

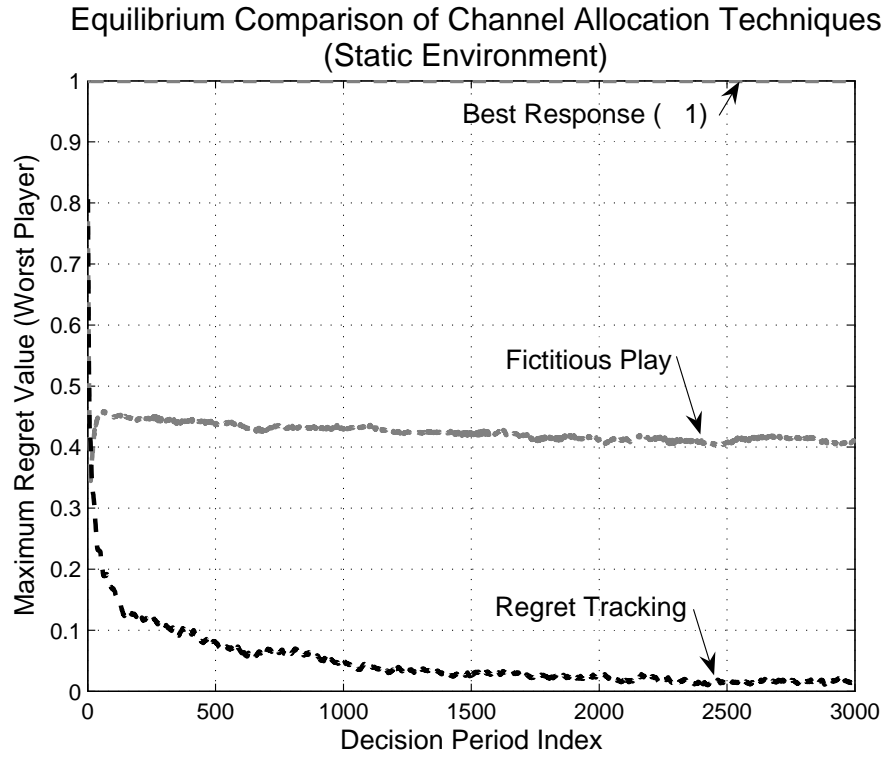


Figure 5.5: Evolution of regret for the channel allocation techniques described in Section 5.5. The regret indicates how close the system is to correlated equilibrium “competitively optimal” behaviour. The modified regret tracking algorithm has large, un-normalized regrets and is not shown.

5.6.3 Performance of Regret Tracking in a Dynamic Environment

Finally, we analyze the performance of Algorithm 5.5.1 when the primary user activity and radio demands vary in time. We first considered $\mathcal{C} = 10$ channels and $L = 6$ users, with uniformly distributed channel qualities and demand levels, and two primary users. At each decision period, each of the demand levels d^l and primary user channel occupation were allowed to change with probability ρ . If a demand level changed, it was recomputed from the uniform distribution; if a primary user changed channels, the new channel was chosen uniformly from among the eight currently unoccupied channels. Next, we analyzed regret tracking in the presence of fast primary users, by allowing each channel quality C_n to be recomputed with probability ρ (slowly varying statistics), with $C_n(i)$ allowed to fluctuate randomly up to $\pm 10\%$ in each decision period (fast varying activity).

Performance of Algorithm 5.5.1 for various values of ρ is shown in Figure 5.6. For w slowly time-varying attributes, the expected time the system remains constant is:

$$T(\rho) = (1 - (1 - \rho)^w)^{-1}. \quad (5.28)$$

We have $w = 16$ with fast primary users, and $w = 8$ without. As expected, average utilization is higher for slower changes, since radios have more time to adapt to their environment. The regret tracking algorithm also outperforms the other algorithms in this case, even when the rate of innovation in the environment is high.

5.7 Conclusions

We have presented a game-theoretic approach for cognitive radio dynamic spectrum access which seeks out and shares temporarily vacant radio spectrum between multiple users. Our approach is completely decentralized in terms of both radio awareness and activity; radios estimate spectral conditions based on their own experience, and adapt by choosing spectral allocations which yield them the greatest utility. Iterated over time, this process converges so that each radio's performance is an optimal response to others' activity. Moreover, we are able to use this apparently selfish scheme to deliver system-wide performance by a judicious choice of utility function.

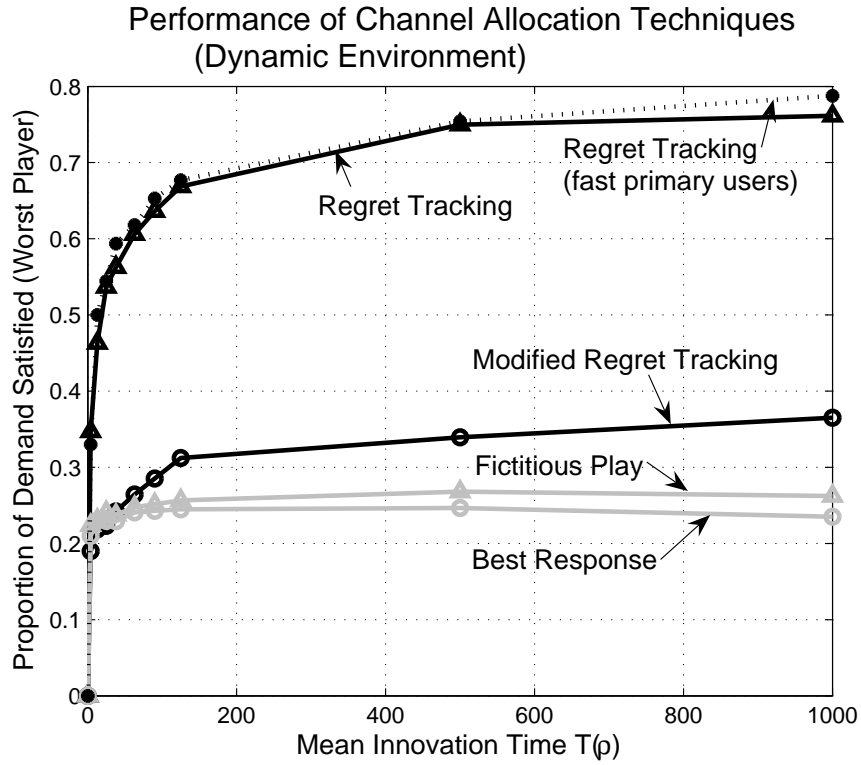


Figure 5.6: Long-run average spectrum utilization in a dynamic environment for the channel allocation techniques of Section 5.5. $T(\rho)$ as in (5.28) is the mean time between innovations in the system (changes in primary user activity, fast primary user statistics or cognitive radio demands).

Bibliography

- [155] R. Aumann. Correlated equilibrium as an expression of bayesian rationality. *Econometrica*, 55(1):1–18, 1987.
- [156] M. Benaim, J. Hofbauer, and S. Sorin. Stochastic approximations and differential inclusions ii: Applications. Levine’s bibliography, UCLA Department of Economics, May 2005.
- [157] U. Berthold and F. Jondral. Guidelines for designing ofdm overlay systems. in *Proc. of the first IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks*, November 2005.
- [158] D. Cabric, S. Mishra, and R. Brodersen. Implementation issues in spectrum sensing for cognitive radios. in *Proc. the 38th. Asilomar Conference on Signals, Systems, and Computers*, pages 772 – 776, 2004.
- [159] K. Challapali, S. Mangold, and Z. Zhong. Spectrum agile radio: detecting spectrum opportunities. in *International Symposium on Advanced Radio Technologies*, 2004.
- [160] Y. Chen, Q. Zhao, and A. Swami. Joint design and separation principle for opportunistic spectrum access. in *IEEE Asilomar Conference on Signals, Systems, and Computers*, 2006.
- [161] Y. Chen, Q. Zhao, and A. Swami. Joint PHY/MAC design of opportunistic spectrum access in the presence of sensing errors. *submitted to IEEE Transactions on Information Theory*, February 2007.
- [162] A. Ghasemi and E. Sousa. Collaborative spectrum sensing for opportunistic access in fading environments. in *Proc. of IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks*, November 2005.
- [163] S. Hart and A. Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.

- [164] S. Hart and A. Mas-Colell. *A reinforcement procedure leading to correlated equilibrium*, in *Economic Essays*, pages 181–200. Springer, 2001.
- [165] S. Hart and A. Mas-Colell. Uncoupled dynamics do not lead to nash equilibrium. *American Economic Review*, 93(5):1830–1836, December 2003.
- [166] S. Haykin. Cognitive radio: Brain-empowered wireless communications. *IEEE Journal on Selected Areas in Communications*, 23(2):201–220, 2005.
- [167] H.J. Kushner and G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer-Verlag, New York, NY, second edition, 2003.
- [168] J. Mitola. Cognitive radio for flexible mobile multimedia communications. *Mob. Netw. Appl.*, 6(5):435–441, 2001.
- [169] N. Nie and C. Comaniciu. Adaptive channel allocation spectrum etiquette for cognitive radio networks. in: *Proc. IEEE DySPAN*, pages 269–278, 2005.
- [170] J. Robinson. An iterative method of solving a game. *Annals of Mathematics*, 54:298–301, 1951.
- [171] S. Sankaranarayanan, P. Papadimitratos, A. Mishra, and S. Hershey. A bandwidth sharing approach to improve licensed spectrum utilization. in: *Proc. IEEE DySPAN*, pages 279–288, 2005.
- [172] W. Wang and X. Liu. List-coloring based channel allocation for open-spectrum wireless networks. in *Proc. of IEEE VTC*, 2005.
- [173] T. Weiss and F. Jondral. Spectrum pooling: an innovative strategy for enhancement of spectrum efficiency. *IEEE Communications Magazine*, pages 8–14, March 2004.
- [174] B. Wild and K. Ramchandran. Detecting primary receivers for cognitive radio applications. November 2005.
- [175] Q. Zhao. Spectrum opportunity and interference constraint in opportunistic spectrum access. in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 2007.
- [176] Q. Zhao and B. Sadler. Dynamic spectrum access: signal processing, networking, and regulatory policy. *IEEE Signal Processing Magazine*, May 2007.

- [177] Q. Zhao, L. Tong, A. Swami, and Y. Chen. Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: a POMDP framework. *IEEE Journal on Selected Areas in Communications: Special Issue on Adaptive, Spectrum Agile and Cognitive Wireless Networks*, April 2007.

Chapter 6

Self-Configuration in Dense Sensor Networks using a Global Game Approach ⁸

6.1 Introduction

Wireless sensor networks, as an emerging technology, promise to achieve high-resolution monitoring in difficult environments by placing sensors up close to the phenomena of interest. To be practical, such networks must be flexible, convenient, energy efficient and as self-regulating as possible. To achieve these goals, it is natural to respond to the *decentralized awareness* of a sensor network with *decentralized information processing* for proper operation. The idea is that if each sensor or small group of sensors can appropriately adapt their behaviour to locally observed conditions, they can quickly self-organize into a functioning network, eliminating the need for difficult and costly central control.

This chapter explores the possibility of using the theory of global games to achieve effective decentralized sensor network operation with minimal communication. Specifically, we devise a global game framework in which each sensor acts as an independent entity to optimize its operation based on environmental signals and predictions about other sensors' behaviour. We consider sensors which can operate in either an energy-efficient "low-resolution" mode, or a more expensive "high-resolution" mode, and seek to configure sensors' strategies for mode selection. This allows us to specify an event-driven framework, adapted to the particular sensor network environment, in which sensors operate at low resolution until a target signal of sufficient intensity is received, at which point high-resolution monitoring and data transmission are initiated. Our main results are:

- Using the theory of global games, we motivate and establish a scenario and conditions

⁸A version of this chapter has been submitted for publication. Maskery, M. and Krishnamurthy, V. Self-Configuration in Dense Sensor Networks using a Global Game Approach, IEEE Transactions on Signal Processing.

under which a simple threshold mode selection strategy provides competitively optimal (Nash equilibrium) behaviour for each sensor. This result holds even when each sensor receives its information in noise. Unlike previous work on global games, we consider the case where both observation noise and utility functions may vary between sensors.

- Given a prior probability distribution for sensor information, which is observed in either Uniform or Gaussian noise, we formulate the Nash equilibrium mode selection thresholds as the solution to set of integral equations. The structure of the solution is examined for limiting low noise conditions, which are particularly important when sensors accumulate and average information over time.
- Equilibrium tests for the threshold strategy are given for both the Uniform and Gaussian noise cases. We also formulate a novel “near equilibrium” condition which provides a simple test for the threshold strategy under low noise conditions.
- A procedure for decentralized computation of the Nash equilibrium threshold mode selection strategy is also given, based on an iterative best strategy response between sensor classes.

The analysis presented is for a one-shot game, that is, the evolution of the sensor environment is not accounted for explicitly, but each sensor faces a static decision problem. These decision problems can, of course, be repeated in time, and we account for this repetition by considering the behaviour of the decision problem as measurements are accumulated, that is as noise is reduced.

The theory of global games was first introduced in [179] as a tool for refining equilibria in economic game theory, see [194] for an excellent exposition. The *global* term refers to the fact that players may actually at any time be playing any game selected from a subclass of all games, which adds an extra dimension to standard gameplay (wherein players act to maximize their own utility in a fixed interactive environment). In a global game, each player observes a noisy signal indicating which of several possible game is being played, given that other players receive similar noisy signals. Since a player’s utility depends on others’ actions, the first step in evaluating alternatives is to predict (conditional on their signal) the actions of others. Global games are attractive, not only because their explicit uncertainty accurately models many real-life situations, but also because their (Nash equilibrium) solution is often easy to characterize and implement. From a game theoretic perspective, our contribution is to extend the work of [187, 194] to the case where players observe their signals under differing, as opposed to identical, noise conditions. While this complicates the analysis,

we are still able to characterize threshold equilibrium strategies where they exist, although players of different classes now have different thresholds. We also find that the equilibrium conditions of [187] can be relaxed without much loss, as long as the amount of noise is small. Another contribution is the use of the well-known Banach fixed point theorem to establish conditions for convergence of the best response dynamic for decentralized computation of the (unique) equilibrium threshold strategy.

Given the tight energy budget of current sensor networks, self-organization and self-configuration are particularly important for efficient operation [178, 181, 193], and have been applied to such diverse problems as routing, topology control, power control, and sensor scheduling. For mode selection, self-configuration allows sensor networks to efficiently extract information, by adjusting individual sensor behaviour to “form to” their environment according to local conditions. Since environments evolve over time, and since centralized organization is costly in terms of communication and energy, self-configuration of sensors is the most feasible method for adapting sensors for this purpose. This leads to robust, scalable and efficient operation since a central authority is not required.

Game theory is a natural tool for describing self-configuration of sensors, since it models each sensor as a self-driving decision maker. A good overview of methods and challenges in this area are given in [183, 190] and the references therein. Despite the practicality and insight provided by global game models, and despite the popularity of game theoretic approaches in the field of sensor networks, we are not aware of any other work applying *global* games in the realm of sensor networks.

Another feature of our approach is that it is event-driven, since it relies on environmental signals for mode selection. This is common for unattended sensor networks for target detection and tracking [182, 186, 189, 191, 192], allowing sensors to act in a decentralized fashion, based on the occurrence of a target event. Moreover, this approach allows us to account for spatial correlation of events and possible contention in the MAC layer to efficiently and reliably deliver information to an end user. The event-driven approach here is closely related to that of [186, 191], in that energy is conserved by realizing that not all sensors need to transmit data to obtain an adequate picture of events.

The rest of this chapter deals with self-configuration and equilibrium operation of sensors through a global game theoretic model for mode selection. Section 6.2 presents the mode selection decision problem as a game. Section 6.3 then motivates and defines a case in which threshold decision rules are in Nash equilibrium (i.e. competitively optimal for each sensor), and gives precise formulations and conditions for threshold equilibrium when noise in the global game is either uniform or Gaussian. Section 6.4 details a decentralized iterative

procedure for self-configuration of thresholds, which converges and considers the limiting case where sensors reduce noise through cumulative averaging of information. Finally, numerical examples and conclusions follow in Sections 6.5 and 6.6.

6.2 A Global Game for Sensor Mode Selection

As mentioned above, global games study the interaction of players who independently choose their actions based on noisy observations of a common signal. This is appropriate for sensor networks, which rely on imperfect information, and do not have unlimited coordination capabilities. In this section we formulate a decision model for sensor mode selection as a global game, with multiple sensor classes. The goal of each sensor is to determine a decision rule for mode selection based on a noisy environmental signal. Since sensor observations are correlated, this signal is also used to predict the actions of other sensors, which is relevant to the decision problem since utilities are interdependent.

6.2.1 Characteristics of Global Games

We begin with a brief discussion of global games. A global games analysis allows us to account for complex sensor behaviour, based on translating noisy environmental observations into predictions of other sensors' behaviour and then into actions. Sensors, imperfectly aware of their environment, choose their best action, assuming that other sensors have information similar to their own. Moreover, each sensor is aware that others are predicting its own behaviour, that they are aware that it is aware, and so on. The *eductive* reasoning process [180, 184] by which sensors iteratively (and without interaction) hypothesize policies and predict reactions to arrive at an optimum, is the justification for the Nash equilibrium conditions considered in this chapter. (It is theoretical only; we favor shortcuts to this iterative process.) The resulting equilibrium has interesting artifacts, such as a bias or clustering effect, in which each sensor (rationally) assumes that its observations are close to the true value, and hence predicts that other sensors will act similarly to itself, even though this may not be the case.

Another commonly studied aspect of global games is strategic supermodularity (complementarity) or submodularity (substitutability) of actions. This is the property that each sensor's incentive to switch to a higher action is increasing (resp. decreasing) in the average action of other sensors. Either property leads to simple, threshold forms for the Nash equilibrium strategy, and we follow [187] in showing how these conditions can be somewhat

relaxed while still retaining the threshold property. This is critical to practical implementation, since simple sensors cannot be expected to execute highly complex policies in reality.

6.2.2 Model Description

In this section we describe the sensor mode selection model in detail, as a global game problem, and characterize the optimal strategy to be implemented by each sensor. We consider a large number N of sensors deployed to watch for events of interest in the field. Our subsequent analysis of the Nash equilibrium is asymptotic as the number of sensors $N \rightarrow \infty$.

Sensor Measurement Model

Each sensor periodically gathers information through either low-resolution or high-resolution measurements, which are periodically transmitted to a base station at a low or high rate, depending on the sensor mode, which is one of two states as chosen by each sensor. The measurements in each mode include determination of an “environmental energy” level X , which indicates the activity level in the local environment, which is measured in noise. The value of measuring in low- or high-resolution mode is dependent on the proportion of other sensors in either mode.

Sensor Types To allow for sensor diversity, we consider multiple *classes* of sensors. We assume that each of the N sensors can be classified into one of I possible classes, where I denotes a positive integer. Let $\mathcal{I} = \{1, 2, \dots, I\}$ denote the set of possible sensor classes, and let N_J be the number of sensors of class $J \in \mathcal{I}$, such that $\sum_{J=1}^I N_J = N$. Furthermore, let $r_J = N_J/N$ denote the proportion of sensors of class $J \in \mathcal{I}$, so $\sum_{J \in \mathcal{I}} r_J = 1$. All sensors of a given class $J \in \mathcal{I}$ will be functionally identical, having the same measurement noise distribution and utility function.

Since the noise distribution and reward parameters will never be known precisely, it is reasonable to perform this type of classification; it is simply a rough division of sensors based on how well they are positioned. Physically, a sensor’s class depends on factors such as the variance of its (low-resolution) measurements, the quality of its high-resolution measurements, and the energy required for it to transmit data (a function of its location in the network).

To fully define the model, we first introduce the following random variables:

- X , the current (true) environmental energy.

- $Y^{(i)}$, Sensor i 's noisy measurement of X .
- $\alpha \in (0, 1)$, the proportion of sensors who respond to their measurement of X by switching to high-resolution mode.

Details of these quantities are given below.

Environmental Energy X : Suppose each sensor continuously monitors for events of interest (e.g. magnetic or mechanical signals with a given signature). For a given space/time window, the actual arrival rate of events is given by a random variable D , with mean m and variance v^2 . If events occur uniformly in space and time, the average arrival rate over M windows is (by the central limit theorem) given by a Gaussian random variable with mean m and variance $s^2 = v^2/M$. It is this value that we denote by the environmental energy X . Note that X is a nonlocal variable, representing an average rate of actual events. We assume that each sensor can obtain an unbiased measurement of X , although the quality of these measurements may vary between sensors due to variable noise conditions.

Noisy Measurement $Y^{(i)}$: Sensor i 's estimate of X can be written as:

$$Y^{(i)} = X + W^{(i)}, \quad (6.1)$$

where $W^{(i)}$ is a random variable representing noise (independent between players). We assume each sensor has a common prior probability distribution for X , and a distribution for $W^{(i)}$, which may vary between sensor classes but not among sensors of a given class. That is, for any class $J \in \mathcal{I}$, for any sensor i of class J , $W^{(i)}$ will have some known distribution ν_J . Details for specific distributions for $W^{(i)}$, (uniform or Gaussian), are given below.

Proportion of “Active” Sensors α : Let α_J represent the proportion of sensors of class $J \in \mathcal{I}$ that are active, that is in high-resolution mode, at a given time. That is

$$\alpha_J = \frac{\# \text{ of active sensors of class } J}{N_J}. \quad (6.2)$$

We then have $\alpha = \sum_{J=1}^I r_J \alpha_J$. Letting $N_J \rightarrow \infty$ for all $J \in \mathcal{I}$ yields a real value for α_J and α . (This is equivalent to assuming a continuum of sensors.)

Reward Function and Policy of each Sensor

The task of the sensor network is ensure that its end users are sufficiently informed of the environment, without expending more effort than required. By considering an autonomous decision model, we are requiring that each sensor supports this goal by independently deciding whether to report its observations or not, based only on $Y^{(i)}$. This reduces costly and complicated active message passing for coordination between sensors. We emphasize that sensors are *never* required to estimate the number of active sensors α ; instead they *predict* this value purely from their observations $Y^{(i)}$. This is a realistic constraint, since estimating α would require multiple rounds of gameplay to enable each sensor to estimate the actions of its counterparts.

However, the autonomous model does not simply imply that a sensor should transmit information whenever its measurement $Y^{(i)}$ is sufficiently large, since it may be better to exploit signal correlation by remaining idle and relying on others to act instead. Indeed, too many active sensors may cause inefficiency due to increased packet collisions (since high-resolution sensors transmit more often), whereas too few may result in insufficient information delivered to the base station. In contrast to the supermodular cases of [180, 194], this gives rise to a game in which the utility is not monotone in α . If α is large, there is a congestion effect which causes the utility to decrease as more sensors switch to high-resolution mode. Fortunately, if the amount of congestion is small, a threshold Nash equilibrium still exists.

To address this, we specify the task of each sensor i is to choose action $u \in S = \{\text{“High_Res”}, \text{“Low_Res”}\}$ (i.e. operate in high resolution or low resolution mode; only transmit data when in high-resolution mode) so as to maximize the expected value of a local reward:

$$C^{(i)}(X, \alpha, u) = \begin{cases} h^{(i)}(X, \alpha), & \text{if } u = \text{“High_Res”} \\ 0, & \text{if } u = \text{“Low_Res”} \end{cases} \quad (6.3)$$

The reward function $h^{(i)}(\cdot)$ is class dependent; for any class $J \in \mathcal{I}$, for any sensor i of class J , $h^{(i)}(\cdot)$ will equal some known function $h_J(\cdot)$ almost everywhere. The reward $h^{(i)}(\cdot)$ is chosen to extract desirable *system* behaviour from optimal *individual* behaviour. In the next section we will specify an exact form, but the essential intuition is that $h^{(i)} = h^{(i)}(\text{“High_Res”}) - h^{(i)}(\text{“Low_Res”})$, where $h^{(i)}(\cdot)$ represents the marginal contribution to some system-wide objective function, minus an energy cost in the appropriate mode.

The goal of sensor i then is to execute a (possibly randomized) strategy $\mu^{(i)} : Y^{(i)} \rightarrow S$ which maximizes $E[C^{(i)}(X, \alpha, \mu)]$, where $\mu = \{\mu^{(i)} : i = 1, 2, \dots\}$ is a collection of strategies

of all sensors. Such a strategy is characterized as follows:

Lemma 6.2.1 $\mu^{(i)}$ is an optimal strategy for Sensor i if

$$\mu^{(i)}(y^{(i)}) = \begin{cases} \text{“High_Res”} & \text{if } E[h^{(i)}(X, \alpha) | Y^{(i)} = y^{(i)}] > 0 \\ \text{“Low_Res”} & \text{if } E[h^{(i)}(X, \alpha) | Y^{(i)} = y^{(i)}] < 0. \end{cases} \quad (6.4)$$

Moreover, $\mu^{(i)}$ is an optimal threshold mode selection strategy if there exists $y^{*(i)}$ such that:

$$E[h^{(i)}(X, \alpha) | Y^{(i)} = y^{(i)}] > 0 \Leftrightarrow y^{(i)} > y^{*(i)}. \quad (6.5)$$

(Proofs are given in Appendix B.) To clarify what is meant by a threshold mode selection strategy (optimal or otherwise), we supply the following definition:

Definition 6.2.1 A threshold strategy $\mu^{(i)}$ is a strategy for sensor i characterized by:

$$\mu^{(i)}(y^{(i)}) = \text{“High_Res”} \Leftrightarrow y^{(i)} > y^{*(i)}, \quad (6.6)$$

where $y^{(i)}$ is a realization of the random observation $Y^{(i)}$, and $y^{*(i)}$ is a specified value characterizing the threshold strategy, referred to as the threshold or switching point. Furthermore, a symmetric threshold strategy is a threshold strategy with the property that for any class $J \in \mathcal{I}$, for any sensor i of class J $y^{*(i)} = y_J^*$.

A block diagram depicting sensor operation is given in Figure 6.1.

In (6.4), a sensor’s optimal strategy depends on the conditional expectation of α , also written $\alpha(\mu)$ to emphasize that the proportion of sensors in “High_Res” mode depends on the sensors’ strategies. This interdependence leads us naturally to seek Nash equilibrium solutions, which is the main goal of the chapter.

6.2.3 Predicting the Proportion of Active Sensors

Implicit in the above formulation, is that each sensor needs to predict the proportion of other sensors selecting each mode, based on its own noisy observations $Y^{(i)}$ of the environment. Assuming that sensors of the same class use the same threshold strategy for mode selection (Symmetric strategies, see Definition 6.2.1), it is possible to write α as a function of X , thus simplifying the threshold computation considerably. As above, let α_J be the proportion of sensors of class $J \in \mathcal{I}$ selecting action “High_Res” (relative to the total number of sensors of that class).

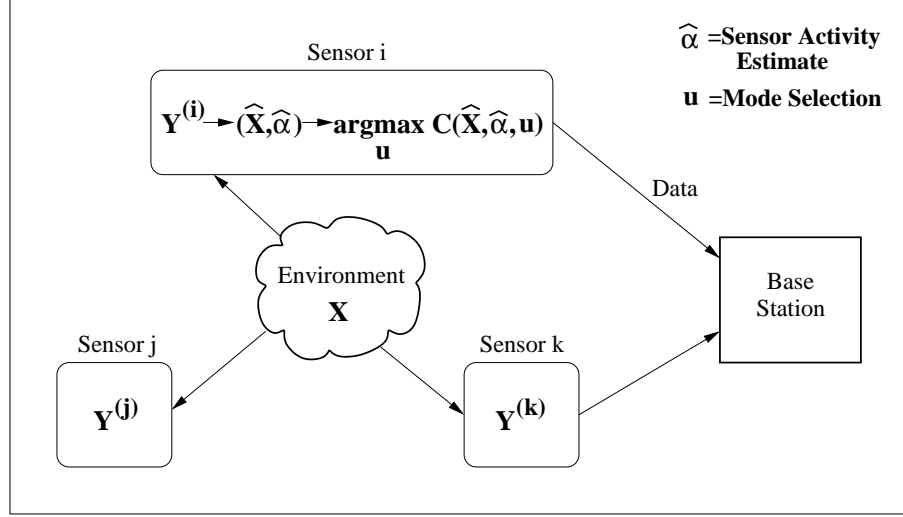


Figure 6.1: Block diagram of sensor mode selection problem. Each sensor observes its environment in noise, then selects its mode based on estimates of the environment and the predicted actions of other sensors.

Since there are N_J total sensors of class J , for symmetric threshold strategy $\mu_J(y_J^*)$, the proportion of sensors that choose “High.Res” is the proportion who receive a signal larger than y_J^* . That is,

$$\alpha_J = \frac{1}{N_J} \sum_{i=1}^{N_J} I\{y^{(i)} > y_J^*\}. \quad (6.7)$$

Taking limits as $N_J \rightarrow \infty$, one obtains $\alpha_J = E[I\{y^{(i)} > y_J^*\}] = Pr(y^{(i)} > y_J^* | X)$. That is:

$$\alpha_J = \alpha_J(X, \mu_J(y_J^*)) \approx \int_{y_J^*}^{\infty} p_{Y^{(i)}|X,J}(y|X) dy, \quad (6.8)$$

and hence,

$$\alpha = \alpha(X, \mu(y^*)) \approx \sum_{J=1}^I r_J \int_{y_J^*}^{\infty} p_{Y^{(i)}|X,J}(y|X) dy. \quad (6.9)$$

Given $X = x$, the probability of $y^{(i)} > y_J^*$ for any sensor i of class J is $\int_{y_J^*}^{\infty} p_{Y^{(i)}|X,J}(y|x) dy$. By the strong law of large numbers this is also (almost surely) the proportion who receive signal $y^{(i)} > y_J^*$.

In the limit, α is thus a deterministic function of X , and for finite numbers of sensors, (6.9) represents an expected proportion of active sensors. In this case, our analysis will be correct under the common notion of “certainty equivalence,” that is, treating the (random) value α as if it were equal to its expected value.

This ability to predict the proportion of active sensors, based only on a noisy signal and some assumptions as to the behaviour of other sensors, is central to the global game approach. It implies that sensors need only measure $Y^{(i)}$, not α , in order to make a mode selection decision. This in turn allows sensors to act simultaneously, without a costly coordination and consultation phase. The conditional prediction of α above will become critical to the threshold levels and Nash equilibrium results in the next section.

6.3 Nash Equilibrium Threshold Strategies

In (6.4), a sensor’s optimal strategy depends on the conditional expectation of α , also written $\alpha(\mu)$ to emphasize that the proportion of sensors in “High_Res” mode depends on the sensors’ strategies. This interdependence leads us naturally to seek Nash equilibrium solutions, defined as follows:

Definition 6.3.1 *A collection of strategies μ is a Nash equilibrium if each $\mu^{(i)}$ is optimal for Sensor i given $\alpha(\mu)$.*

The rest of this chapter will explore Nash equilibrium strategies in detail, focusing on criteria for the existence of **threshold** Nash equilibria. Such strategies are straightforward to characterize and simple to implement, since sensors need only follow a simple threshold rule for mode selection.

Optimal strategy μ depends on the expected behaviour of the other sensors’ behaviour through α , the total proportion of sensors choosing action “High_Res.” This in turn depends on the strategy of the other sensors. We are therefore interested in determining a collection of strategies for each sensor that are simultaneously optimal, i.e. in Nash equilibrium. For simplicity, we concentrate on collections of strategies μ comprised of symmetric threshold strategies, as in Definition 6.2.1.

Such strategies are attractive because they are practical for implementation in simple, inexpensive sensors, since only a single parameter y_J^* need be computed or specified for each sensor class $J \in \mathcal{I}$. We are therefore interested in conditions under which satisfactory performance (i.e. Nash equilibrium) can be obtained by such collections of symmetric threshold strategies. In this section we formulate potential Nash equilibrium threshold strategies for

sensors under differing noise conditions (Uniform or Gaussian), and give conditions under which the proposed threshold strategies are in Nash equilibrium.

It is straightforward to show that the conditional expectation in Lemma 6.2.1 is continuous in the signal $y^{(i)}$. Hence, it is reasonable to propose as a set of mode selection thresholds the solution to:

$$\left\{ y_J^* : E[h_J(X, \alpha) | Y^{(i)} = y_J^*] = 0 : J \in \mathcal{I} \right\}. \quad (6.10)$$

The corresponding threshold strategy is then optimal for any sensor i of class J if:

$$E[h_J(X, \alpha) | Y^{(i)} = y^{(i)}] > 0 \Leftrightarrow y^{(i)} > y_J^*. \quad (6.11)$$

If the collection (6.10) is optimal for every sensor, then a symmetric threshold Nash equilibrium has obtained, that is, the collection of symmetric threshold strategies as in Definition 6.2.1 is in Nash equilibrium. We will explore specific conditions under which this holds in Section 6.3.

To motivate the existence of Nash equilibrium threshold strategies for the mode selection problem, suppose the system (global) utility of information gathered when proportion α of sensors are in mode “High.Res”, given X is:

$$H(X, \alpha) = X\alpha + G_s(\alpha). \quad (6.12)$$

The implication here is that higher environmental energy should elicit more high-resolution monitoring from the network, but this trend is perturbed by secondary nonlinear effects, given by $G_s(\alpha)$.

Our goal is to maximize the global utility subject to energy constraints in a decentralized manner. Thus, each sensor is equipped with its own utility as follows. Let the additional energy required by Sensor i in high-resolution mode be proportional to $G_e^{(i)}(\alpha)$, and let $c^{(i)}$ represent the quality of information obtained by i . Then if we take i 's reward $h^{(i)}$ to be its marginal contribution to the system utility, less its energy cost, we may write:

$$h^{(i)}(X, \alpha) = c^{(i)} \frac{\partial H}{\partial \alpha}(X, \alpha) - G_e^{(i)}(\alpha) = c^{(i)} X + f^{(i)}(\alpha), \quad (6.13)$$

$$\text{where } f^{(i)}(\alpha) = c^{(i)} G_s'(\alpha) - G_e^{(i)}(\alpha). \quad (6.14)$$

A practical implementation of such a utility function is given in [191] in the context of Zigbee-enabled unattended ground sensor networks. If the impact of $f^{(i)}(\alpha)$ is small enough, it is reasonable to expect that a threshold strategy will be optimal, since for high values of

X , (6.13) will be positive, and hence “High_Res” will be the optimal action. Based on this intuition, we focus on threshold strategies for the remainder of the chapter.

Using (6.13) and (6.9), we may now quantify the conditional expectation of reward h_J as:

$$\begin{aligned} E[h_J(X, \alpha) | Y^{(i)} = y^{(i)}] \\ = c_J E[X | Y^i = y^{(i)}] + \int_{-\infty}^{\infty} f_J \left(\sum_{k \in I} r_k \int_{y_k^*}^{\infty} p_{Y^{(i)} | X, k}(y | x) dy \right) p_{X | Y^{(i)}, J}(x | y^{(i)}) dx. \end{aligned} \quad (6.15)$$

Note there is a *cluster effect* here: sensors with a higher signal $Y^{(i)}$ tend to have a higher expectations of X and hence predict a higher proportion α of active sensors. This subtle bias must be accounted for in the Nash equilibrium.

In the remainder of this section, we characterize the threshold strategies and give conditions under which the resulting strategy is a Nash equilibrium. We consider two different cases for X and $W^{(i)}$. In Section 6.3.1 we study the case when the prior on X is diffuse and $W^{(i)}$ is uniformly distributed; in Section 6.3.2 we study the case when both X and $W^{(i)}$ have Gaussian distributions.

6.3.1 Nash Equilibrium Threshold Strategies for Uniform Observation Noise

This section develops the Nash equilibrium threshold strategy for the case where there is no prior information on X , which is observed in uniform noise. If sensors are deployed in an unknown environment, there may be no prior knowledge of the distribution of environmental energy X , and the distribution of observation noise may be unknown. In this case, one may take a conservative approach, by supposing that:

$$X \sim \text{Uniform}(-\infty, \infty), \quad W^{(i)} \sim \text{Uniform}(-\varepsilon_J, \varepsilon_J),$$

where Sensor i is of class J . This approach is taken, in the context of players who must decide whether to go to a nightclub (given the quality of music and having decreased utility if the club is either too empty or too crowded) in [187]. However, that work only allows for one sensor class; we extend the results to multiple classes here. The diffuse nature of the prior distribution does not cause any technical difficulties here, since we are only interested in the posterior distribution on X given $Y^{(i)}$, which is well-defined [185].

The value and equilibrium conditions for threshold strategies in this case are fairly simple,

since the uniform distribution has finite support. The thresholds are given as the solution to a system of integral equations, which represents an interesting extension to the known single-class case for global games with congestion. The main result, giving the general form of the threshold and equilibrium conditions, is as follows:

Theorem 6.3.1 1.) *For uniform noise and diffuse prior, the set of switching points of the symmetric threshold strategy (Definition 6.2.1) in the global game must satisfy the system:*

$$\left\{ y_J^* = -\frac{1}{2\varepsilon_J c_J} \int_{y_J^* - \varepsilon_J}^{y_J^* + \varepsilon_J} f_J \left(\sum_{k \in \mathcal{I}} r_k \left[\frac{x - y_k^* + \varepsilon_k}{2\varepsilon_k} \right]_0^1 \right) dx : J \in \mathcal{I} \right\}. \quad (6.16)$$

That is, in Nash equilibrium, the threshold strategy must switch at y_J^ ; if signal $y^{(i)} < y_J^*$ for a sensor of class J , then the sensor should select “Low_Res” mode, otherwise it should select “High_Res” mode. 2.) Furthermore, the collection of all threshold strategies is a Nash equilibrium if:*

$$f_J \left(\sum_{k \in \mathcal{I}} r_k \left[\frac{y^{(i)} - \varepsilon_J - y_k^* + \varepsilon_k}{2\varepsilon_k} \right]_0^1 \right) - f_J \left(\sum_{k \in \mathcal{I}} r_k \left[\frac{y^{(i)} + \varepsilon_J - y_k^* + \varepsilon_k}{2\varepsilon_k} \right]_0^1 \right) \leq 2\varepsilon_J c_J \quad \forall y^{(i)} \quad (6.17)$$

for all $J \in \mathcal{I}$, and the collection of all threshold strategies is not a Nash equilibrium if:

$$f_J \left(\sum_{k \in \mathcal{I}} r_k \left[\frac{y_J^* - \varepsilon_J - y_k^* + \varepsilon_k}{2\varepsilon_k} \right]_0^1 \right) - f_J \left(\sum_{k \in \mathcal{I}} r_k \left[\frac{y_J^* + \varepsilon_J - y_k^* + \varepsilon_k}{2\varepsilon_k} \right]_0^1 \right) > 2\varepsilon_J c_J \quad (6.18)$$

for any $J \in \mathcal{I}$.

The notation $[x]_0^1$ denotes the operation $\min\{1, \max\{0, x\}\}$. The threshold strategy switching point above is essentially an indifference point; if a sensor receives a signal $Y^{(i)} = y_J^*$, then it will be indifferent between its two modes, since it expects to receive an expected reward of zero either way. The equilibrium condition is really that this point be the only zero-crossing of the expected reward $E[h(X, \alpha | Y^{(i)} = y^{(i)})]$ as a function of $y^{(i)}$. Solutions to this nonlinear system may be easily obtained numerically, e.g. by Newton’s method, or by online adaptive computation, as in the discussion on self-configuration in Section 6.4.1. Note that if ε_J is large, then (6.17) is always satisfied if f_J is bounded.

To gain some intuition and connection with other global game results, suppose the index set \mathcal{I} is rearranged so that $y_1^* < y_2^* < \dots$, and consider the special “disjoint switching” case where $y_J^* + \varepsilon_J \leq y_k^* - \varepsilon_k$ for all $J \neq k$. That is, when the observation noise ε_k becomes small

for all classes $k \in \mathcal{I}$. This leads to the following corollary:

Corollary 6.3.1 *Define $R_J = \sum_{k \leq J} r_k$, (where r_k is the proportion of all sensors of class $k \in \mathcal{I}$) and assume the switching points of the threshold strategies satisfy $y_1^* < y_2^* < \dots$ and $y_J^* + \varepsilon_J \leq y_k^* - \varepsilon_k$ for all $J \neq k$. We obtain for (6.16):*

$$y_J^* = -\frac{1}{c_J r_J} \int_{R_{J-1}}^{R_J} f_J(\alpha) d\alpha = -\frac{1}{c_J} \int_{R_{J-1}/r_J}^{R_J/r_J} f_J(r_J \tau) d\tau. \quad (6.19)$$

Corollary 6.3.1 reveals a novel result: the “multi-class Laplacian” condition for mode selection. This is analogous to the single-class version, detailed in [187, 194], which can be recovered by considering just one class J (and absorbing c_J into f_J), yielding $y^* = -\int_0^1 f(\alpha) d\alpha$. This implies that the mode selection threshold could be equivalently obtained by simply assuming that the proportion α of high resolution sensors is uniformly distributed over the interval $[0, 1]$. In the case of multiple classes (in the disjoint switching case), the assumption is that the proportion α_J of high resolution sensors of ones own class are uniformly distributed, while sensors of other classes are either all active ($k < J$) or all inactive ($k > J$).

Corollary 6.3.2 *For a sensor network consisting of a single sensor class (i.e. $\mathcal{I} = \{1\}$), (6.17) reduces to:*

$$f\left(\left[\frac{y^{(i)} - y^*}{2\varepsilon}\right]_0^1\right) - f\left(\left[\frac{y^{(i)} - y^*}{2\varepsilon} + 1\right]_0^1\right) \leq 2\varepsilon \quad \forall y^{(i)}, \quad (6.20)$$

or equivalently,

$$f(0) - \underline{f} \leq 2\varepsilon, \text{ and } \bar{f} - f(1) \leq 2\varepsilon. \quad (6.21)$$

and (6.18) becomes:

$$f(0) - f(1) > 2\varepsilon. \quad (6.22)$$

These are very similar to the conditions in Remarks 1 and 2 of [187], but more general, as we do not require $f(0) = 0$ or quasi-concavity. The interpretation there (and here) is that f cannot fall too much from its beginning to its minimum value, or from its maximum to its ending value. In the words of [187], there cannot be too much “congestion” in the system, relative to the noise. Conditions (6.17) and (6.18) essentially require the same property, but the overlap of classes makes the role of congestion more subtle.

6.3.2 Threshold Strategies for Gaussian Priors and Noise

This section develops the Nash equilibrium threshold strategy for the case where the prior information on X , and the observation noise are both characterized by Gaussian distributions. In many applications, more prior information about X and $W^{(i)}$ will be available, which sensors may incorporate into their computations. In this section, we repeat the analysis of Section 6.3.1 for Gaussian conditions, which are commonly encountered in practice. For sensor i of class J , assume:

$$X \sim N(m, s^2) \quad , \quad W^{(i)} \sim N(0, \sigma_J^2).$$

Then by well-known properties of the Gaussian distribution, we obtain:

$$Y^{(i)} \sim N(m, s^2 + \sigma_J^2), \quad Y^{(i)} | (X = x) \sim N(x, \sigma_J^2), \quad X | (Y^{(i)} = y^{(i)}) \sim N\left(\nu_J^{(i)}, \omega_J^2\right),$$

$$\text{where } \nu_J^{(i)} = \frac{\sigma_J^2}{s^2 + \sigma_J^2} m + \frac{s^2}{s^2 + \sigma_J^2} y^{(i)}, \quad \omega_J^2 = \frac{s^2 \sigma_J^2}{s^2 + \sigma_J^2}. \quad (6.23)$$

The threshold values and equilibrium conditions for this case are given by the following theorem:

Theorem 6.3.2 *Let*

$$\Psi_J(y, z) = \sum_{m \in \mathcal{I}} r_m Q\left(\frac{y_m^* - \omega_J z - \nu_J(y)}{\sigma_m}\right). \quad (6.24)$$

Then,

1. *For Gaussian noise and prior, the set of switching points of the threshold strategy (Definition 6.2.1) in the global game must satisfy the system:*

$$\left\{ y_J^* = -\frac{\sigma_J^2 m}{s^2} - \frac{(s^2 + \sigma_J^2)}{c_J s^2} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f_J(\Psi_J(y^*, z)) e^{-z^2/2} dz : J = \mathcal{I} \right\}. \quad (6.25)$$

2. *Furthermore, the collection of all threshold strategies is a Nash equilibrium if:*

$$-\frac{1}{c_J} \sum_{k \in \mathcal{I}} \frac{r_k}{\sqrt{2\pi} \sigma_k} \int_{-\infty}^{\infty} f'_J(\Psi_J(y, z)) e^{-\left(\frac{y_k^* - \omega_J z - \nu_J(y)}{\sigma_k}\right)^2 / 2} e^{-z^2/2} dz < 1, \quad (6.26)$$

for all y and all $J \in \mathcal{I}$, and the collection of all threshold strategies is not a Nash

equilibrium if:

$$-\frac{1}{c_J} \sum_{k \in \mathcal{I}} \frac{r_k}{\sqrt{2\pi}\sigma_k} \int_{-\infty}^{\infty} f'_J(\Psi_J(y, z)) e^{-\left(\frac{y_k^* - \omega_J z - \nu_J(y_J^*)}{\sigma_k}\right)^2/2} e^{-z^2/2} dz > 1, \quad (6.27)$$

for any $J \in \mathcal{I}$.

Corollary 6.3.3 *As in the uniform case, suppose the index set \mathcal{I} of sensor classes is rearranged so that $y_1^* < y_2^* < \dots$, and consider the limiting case in which $\sigma_k^2 \rightarrow 0$ for all $k \in \mathcal{I}$, and such that $\sigma_J/\sigma_k < \infty$ for all $J, k \in \mathcal{I}$. Then for each $J \in \mathcal{I}$, the threshold point satisfies:*

$$y_J^* \rightarrow -\frac{1}{c_J \sqrt{2\pi}} \int_{-\infty}^{\infty} f_J(R_{J-1} + r_J Q(-z)) e^{-z^2/2} dz = -\frac{1}{c_J r_J} \int_{R_{J-1}}^{R_J} f_J(\alpha) d\alpha, \quad (6.28)$$

where $R_J = \sum_{k \leq J} r_k$.

This is exactly the same “multi-class Laplacian” condition as in the uniform case [187, 194], which is intuitive since the two probability models converge as noise is decreased.

Unfortunately, the derivative condition, especially (6.26), does not easily yield a simple closed-form expression here as in the uniform case. Clearly, however, if $f'_J > 0$ everywhere, then the threshold strategy for mode selection is a Nash equilibrium. Moreover, if σ_k is large, then (6.26) is always satisfied. The equilibrium condition (6.27) for the small-noise case is given as follows:

Corollary 6.3.4 *If σ_k is small for all k , then $\mu(y^*)$ is not an equilibrium if:*

$$\sigma_J < \frac{-1}{c_J} \int_{R_{J-1}}^{R_J} f'_J(\alpha) d\alpha, \text{ for any } J \in \mathcal{I}. \quad (6.29)$$

We will derive more tractable conditions in Section 6.4.2 for the case when σ_k is small.

6.4 Further Analysis of Mode Selection Thresholds

In this section, we consider some practicalities involved with implementing threshold strategies for a wireless sensor network. First, the problem of decentralized, self-configuration of the threshold strategies is addressed in Section 6.4.1 by an iterative algorithm for threshold computation. Next, hypothesizing that sensors may reduce their noise by averaging their accumulated history of measurements, we examine the robustness of the equilibrium property of the threshold strategy. This results in an interesting “near” equilibrium condition

which can be easily checked to determine whether threshold operation is appropriate for a given scenario.

6.4.1 Adaptive Self-Configuration of Mode Selection Thresholds

For a self-configuring sensor network, it is necessary that each sensor determines its strategy without supervision. Since we are concerned with symmetric threshold strategies in this chapter, we outline in this section an iterative method for computing a set of thresholds for mode selection that are a candidate Nash equilibrium. The threshold depends only on the sensor's class, and the proportion of other classes (locally) present. Since classes change infrequently (they are the parameters of the system), it is reasonable to determine thresholds iteratively through repeated interaction.

We propose the following iterative procedure for computing the set of thresholds $\{y_J^* : J \in \mathcal{I}\}$:

Algorithm 6.4.1 Dynamic Threshold Configuration:

1. Each sensor of class J (or group of sensors) proposes an initial threshold $y_J^*(0)$.
2. For $n = 1, 2, \dots$ until $y_J^*(n) \approx y_J^*(n-1)$, repeat:

$$y_J^*(n) = z : E[h_J(X, \alpha(X, \mu(y^*(n-1)))) | Y^{(i)} = z] = 0. \quad (6.30)$$

Referring to (6.16) and (6.25), this procedure can be written as the iteration of a map $y^*(n) = T(y^*(n-1))$, where T has components:

$$T_J(y^*) = -\frac{1}{2\varepsilon_J c_J} \int_{y_J^* - \varepsilon_J}^{y_J^* + \varepsilon_J} f_J \left(\sum_{k \in \mathcal{I}} r_k \left[\frac{x - y_k^* + \varepsilon_k}{2\varepsilon_k} \right]_0^1 \right) dx \quad (6.31)$$

for the uniform case, and

$$T_J(y^*) = -\frac{\sigma_J^2 m}{s^2} - \frac{(s^2 + \sigma_J^2)}{c_J s^2} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f_J \left(\sum_{k \in \mathcal{I}} r_k Q \left(\frac{y_k^* - \omega_J z - \nu_J^*}{\sigma_k} \right) \right) e^{-z^2/2} dz \quad (6.32)$$

for the Gaussian case.

By Banach's fixed point theorem [188], if T is a contraction, then the iterative computation will converge. Moreover, the limit point is unique; it is the only point satisfying $y^* = T(y^*)$. The fixed point theorem in this case is stated as follows:

Theorem 6.4.1 *Suppose T is a contraction, that is there exists a $q < 1$ such that (using the sup-norm):*

$$\sup_{J \in \mathcal{I}} |T(y) - T(w)| \leq \sup_{J \in \mathcal{I}} q |y - w| \text{ for all } y, w \in \mathbb{R}^{\mathcal{I}}.$$

Then Algorithm 6.4.1 converges to a unique threshold y^ . Furthermore, convergence is exponentially fast, in the sense that:*

$$\|T(y^*) - T(y^*(n))\| \leq \frac{q^n}{1 - q} \|T(y^*(1)) - T(y^*(0))\|. \quad (6.33)$$

Theorem 6.4.1 is useful because it significantly relaxes the conditions for the iterative procedure; proposals do not have to be universally circulated, or even communicated at all. Small groups or individual sensors can compute the thresholds so as to avoid excessive communication costs. In practice, (see Section 6.5.3) we found that T was a contraction for a wide range of examples.

6.4.2 Robustness of Threshold under Measurement Averaging

As measurements are accumulated by sensors, they may be averaged to produce a better estimate for X . It is therefore natural that there will be less uncertainty as time goes on, as represented by reduced noise σ_J^2 or ε_J . The purpose of this section is to determine when the threshold strategy $\mu(y^*)$ remains a Nash equilibrium as sensors gain more information about their environment.

In the Gaussian case, it is easy to see how noise in $W^{(i)}$ may decrease in time, by simply taking a time average of past measurements. That is, take

$$Y^{(i)} = X + \frac{1}{N} \sum_{n=1}^N W_n^{(i)} = X + \bar{W}^{(i)}. \quad (6.34)$$

(If X evolves according to known dynamics, this can be generalized using a Kalman filtering approach.) By the central limit theorem, the variance of $\bar{W}^{(i)}$ is reduced to σ_J^2/N . In the uniform case, average measurements also lead to a Gaussian distribution for $\bar{W}^{(i)}$ by the central limit theorem.

For low noise conditions, there is a simple necessary and sufficient condition for $\mu(y^*)$ to be “nearly” a Nash equilibrium. That is, as long as the noise value is less than a given maximum value, and as long as some simple conditions are satisfied, it is optimal for Sensor i of class J to follow threshold strategy $\mu_J(y_J^*)$ except for signal $y^{(i)}$ on a very narrow interval. Moreover, y_J^* is particularly simple to compute in this case, as we saw in (6.19)

and (6.28). Our result is stated as follows (for completeness we consider both uniform and Gaussian noise, although only Gaussian is required in the motivation above):

Theorem 6.4.2 *For each sensor i of class $J \in \mathcal{I}$, suppose ε_J (uniform noise, Section 6.3.1) or σ_J (Gaussian noise, Section 6.3.2) are small, and σ_J/σ_k is near unity for all $J, k \in \mathcal{I}$. Then, the collection of threshold strategies of all sensors, denoted $\mu(y^*)$, is a Nash equilibrium everywhere except possibly on the small region where any Sensor i receives a signal $y^{(i)}$ “close to” y_k^* for some $k \in \mathcal{I}$, if and only if the following conditions on the measurement at each sensor i holds:*

$$\begin{aligned} \text{For all } J \in \mathcal{I}, y^{(i)} > y_J^* &\Rightarrow y^{(i)} > \frac{-1}{c_J} f_J \left(\sum_{k \in \mathcal{I}} r_k I\{y^{(i)} > y_k^*\} \right), \text{ and,} \\ y^{(i)} < y_J^* &\Rightarrow y^{(i)} < \frac{-1}{c_J} f_J \left(\sum_{k \in \mathcal{I}} r_k I\{y^{(i)} > y_k^*\} \right), \end{aligned} \quad (6.35)$$

where c_J, f_J are utility parameters as in (6.13), (6.14), and r_J is the proportion of sensors of class J , for $J \in \mathcal{I}$. The small region where following the threshold strategy may not be optimal is given by:

$$y^{(i)} : |y^{(i)} - y_k^*| > \varepsilon_J + \varepsilon_k \text{ for some } k \in \mathcal{I}, \text{ (uniform) or,} \quad (6.36)$$

$$y^{(i)} : |y^{(i)} - y_k^*| > 3(\sigma_J + \sigma_k) \text{ for some } k \in \mathcal{I}, \text{ (Gaussian)} \quad (6.37)$$

for sensor i of class J .

The advantage here is that the right-hand side of (6.35) is piecewise constant, rendering the expected utility computation particularly simple. This allows us to specify the necessary and sufficient condition for the threshold strategy $\mu(y^*)$ to be “nearly” an equilibrium.

When only a single sensor class is present (i.e. $\mathcal{I} = \{1\}$), it is straightforward to show Theorem 6.4.2 is equivalent to the following:

Corollary 6.4.1 *Suppose there is one class of sensor, with small ε or σ . Then the threshold strategy $\mu(y^*)$ for mode selection is a Nash equilibrium except possibly for:*

$$y^{(i)} : |y^{(i)} - y_k^*| > 2\varepsilon, \text{ or, } y^{(i)} : |y^{(i)} - y_k^*| > 6\sigma, \quad (6.38)$$

if and only if:

$$f(0) < \int_0^1 f(\alpha) d\alpha < f(1). \quad (6.39)$$

Theorem 6.4.2 and Corollary 6.4.1 give simple conditions under which the threshold strategy remains a near-Nash equilibrium for small noise conditions. In design, this is useful for determining system parameters for which the threshold strategy is a good choice under any noise conditions. If the conditions are not satisfied, a closer look at the anticipated noise conditions should reveal whether or not such a strategy is appropriate.

This result should be contrasted with those of [187, 194], which suggest that optimality of the Nash equilibrium is *not* preserved in the limit. Strictly speaking, the same is true in our case, for example, as $\varepsilon_J \rightarrow 0$ equilibrium conditions (6.17), (6.18) converge to a condition:

$$\mu_J(y_J^*) \text{ is in equilibrium} \Leftrightarrow f_J(R_{J-1}) < f_J(R_J). \quad (6.40)$$

That is, f_J must be increasing at discrete points; if we allow arbitrary r_J , f_J must be increasing everywhere. Thus, any amount of congestion (negative slope) will eventually lead to suboptimality of the threshold strategy. However, our point is that for low noise conditions, this should not matter since the signal $y^{(i)}$ will rarely fall there. Moreover, in the limit as the noise goes to zero, these regions vanish altogether.

If the equilibrium condition of Theorem 6.4.2 fails, we can still use Corollary 6.3.4 to say exactly when the threshold fails. Specifically, when N measurements have been taken such that:

$$\frac{\sigma_J}{\sqrt{N}} < \frac{-1}{c_J} \int_{R_{J-1}}^{R_J} f'_J(\alpha) d\alpha, \text{ for any } J \in \mathcal{I}, \quad (6.41)$$

then the threshold strategy $\mu(y^*)$ can no longer be a Nash equilibrium. This is again a consequence of high congestion, since (6.41) is more quickly satisfied when f_J is decreasing quickly on average over the given region.

We conclude with a few words about non-threshold Nash equilibria. If the conditions above are not met, Nash equilibria for the global game still exist, but are highly complex. In [187] it was suggested that each individual player could have a different strategy in equilibrium, consisting of disjoint signal regions over which the optimal action alternates between “Low_Res” and “High_Res.” These strategies are deterministic (by Lemma 6.2.1), but inherently unpredictable. Such erratic behaviour is not desirable in sensor networks, nor are the equilibria easy to compute, and we suggest that it should prompt further design measures to alleviate congestion in the game.

6.4.3 Robustness of Threshold under Variance-based Measurement Quality

In this section we suppose that the environmental energy X is the key characteristic to be probed by the sensor network. Instead of averaging multiple measurements as in Section 6.4.2, one can use previous measurements to reduce uncertainty in the prior distribution of X . In addition, the quality-of-information weighting $c^{(i)}$ in (6.13) can be tied (in the Gaussian case) to the ratio between observation noise and uncertainty. The latter is reflected by taking:

$$c^{(i)} = d^{(i)} \frac{s^2}{\sigma^{(i)2}}, \quad (6.42)$$

for some utility parameter $d^{(i)}$. This affects the utility by which sensors take decisions:

$$h^{(i)}(X, \alpha) = d^{(i)} \frac{s^2}{\sigma^{(i)2}} X + d^{(i)} \frac{s^2}{\sigma^{(i)2}} G'_s(\alpha) - G_e(\alpha). \quad (6.43)$$

In this section we examine behaviour of the threshold and Nash equilibrium under this reformulation, as the variance s^2 on the prior distribution of X evolves under successive observations.

In general, if the “environmental energy” X evolves according to a linear system, taking value X_n at discrete time n , Sensor i can use a Kalman filter to deterministically update the variance $s_n^{(i)2}$ of X_n over time. For the static case, we have by (6.23) that the sequence $s_1^{(i)2}, s_2^{(i)2}, \dots$ satisfies:

$$s_n^{(i)2} = \frac{s_{n-1}^{(i)2} \sigma^{(i)2}}{s_{n-1}^{(i)2} + \sigma^{(i)2}}. \quad (6.44)$$

Clearly then $\lim_{n \rightarrow \infty} s_n^{(i)2} = 0$.

Under this reformulation, the following result holds:

Lemma 6.4.1 *The threshold points $y_J^* \rightarrow \infty$. Furthermore, the threshold strategy cannot be an equilibrium if $G'_e(\alpha) > 0$ everywhere.*

The proof of Lemma 6.4.1 follows straightforwardly from substitution of (6.43) and letting $s^{(i)2} \rightarrow 0$.

Lemma 6.4.1 holds even in the limit, as $\sigma_k^2 \rightarrow 0$ for all k , and such that $\sigma_J/\sigma_k < \infty$ everywhere. Note that $G'_e(\alpha) > 0$ is a more or less standard assumption, since greater congestion is likely to lead to more energy required for transmission.

Lemma 6.4.1 shows that, if sensors price their information according to the ratio between prior uncertainty and observation uncertainty, then initially the threshold may be low, but

as prior uncertainty decreases, the threshold will increase until all sensors remain in their low-power, low resolution mode. At this point there is simply no incentive to make further costly measurements, since X is already known with near certainty.

6.5 Numerical Examples

We have shown that for large noise conditions, $\mu(y^*)$ is a Nash equilibrium, and for small noise conditions, a simple test can be performed to check whether this is the case. To gain some intuition as to when the equilibrium holds at intermediate values, we turn to numerical examples. We begin by analyzing a sensor network comprising of a single class of sensor and weak congestion effects, which is then extended to multiple sensor classes and to stronger congestion, which eventually destroys the equilibrium property of the threshold strategy for mode selection. We conclude by investigating the convergence properties of the iterative procedure for threshold computation in Section 6.4.1.

6.5.1 Single Utility, Weak Congestion

To begin, we assume each sensor i has identical utility parameters:

$$c^{(i)} = 0.05, \quad f^{(i)}(\alpha) = -0.4106\alpha^2 + 0.5517\alpha + 0.0301,$$

see Figure 6.2. In this case, $f^{(i)}$ decreases when the system becomes crowded (large α) but this congestion is never sufficient to switch any sensor back to its low-resolution mode. In fact, calculating $f(0) = -0.03$, $\int_0^1 f(\alpha)d\alpha = 0.1089$, and $f(1) = 0.111$, we see that Corollary 6.4.1 is satisfied, so the threshold strategy $\mu(y^*)$ will always be a Nash equilibrium. The threshold value is constant for the uniform noise case at $y^* = -2.178$, which is also the limiting threshold value for small noise conditions in the Gaussian case.

Figure 6.3 shows the single crossing property of $E[h^{(i)}(X, \alpha) | Y^{(i)} = y^{(i)}]$, which ensures the threshold strategy is an equilibrium. That is, it is optimal for each sensor to select “High_Res” mode if $y^{(i)} > y^*$, since the expected reward is positive in this case, and for each sensor to select “Low_Res” mode if $y^{(i)} < y^*$, where the expected reward is negative. This property is preserved even when the noise becomes small. The figure is shown for a Gaussian prior on X with mean $m = 2$ and variance $s^2 = 16$, and several Gaussian noise distributions. Note how the threshold y^* drifts toward the limiting value as σ^2 decreases, according to Theorem 6.3.2. The uniform noise case behaves similarly, except that the threshold remains constant.

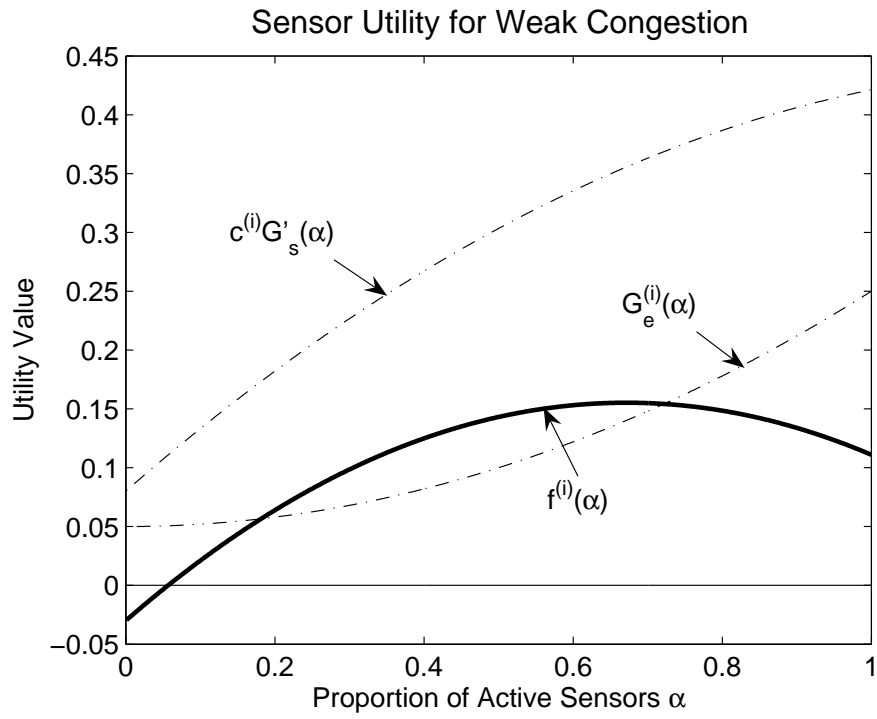


Figure 6.2: Sensor utility function for first example. The interactive component $f^{(i)}(\alpha)$ is the difference between the expected contribution to the end-user's information, $c^{(i)}G'_s(\alpha)$ and the energy cost, $G_e^{(i)}(\alpha)$, as in (6.14).

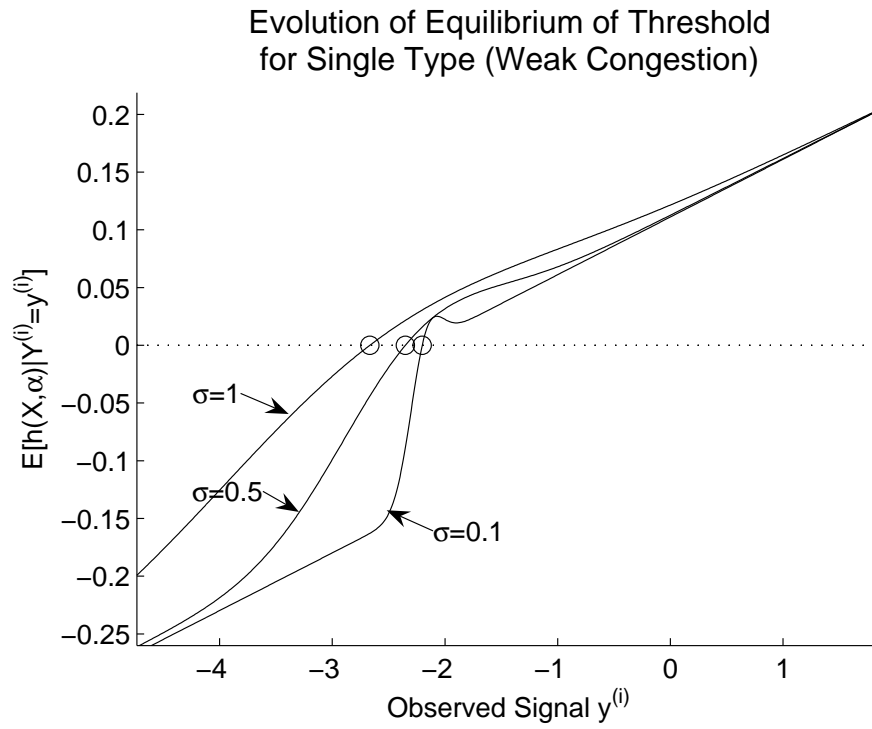


Figure 6.3: As the observation noise σ decreases, the threshold y^* increases, and more fluctuation is observed in the expected utility. However, the utility function is such that there is a single crossing as $\sigma \rightarrow 0$, so the threshold strategy is a Nash equilibrium.

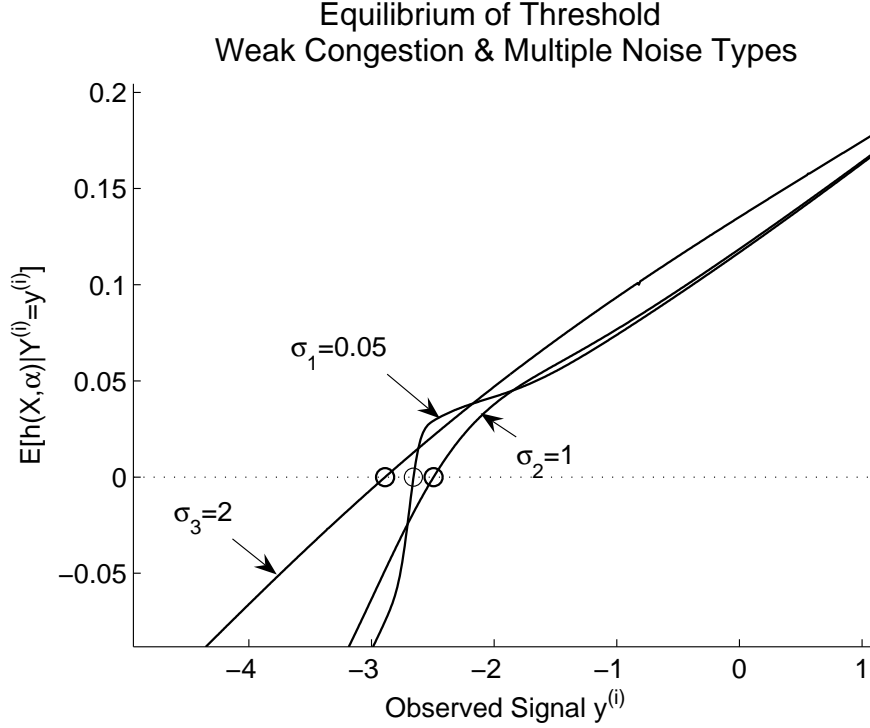


Figure 6.4: Example of the expected utility and thresholds (zero-crossing points) for multiple sensor classes. Lower noise corresponds to more fluctuation in expected utility, but does not necessarily result in a lower threshold.

A similar result holds when sensors have identical utility parameters, but are separated into multiple classes based on their noise variances $\sigma^{(i)}$. In Figure 6.4, we show the values of $E[h^{(i)}(X, \alpha) | Y^{(i)} = y^{(i)}]$, for three noise classes: $\sigma_1 = .05, \sigma_2 = .5, \sigma_3 = 2$, with $r_1 = 0.4, r_2 = r_3 = 0.3$. These result in slightly different thresholds $\{y_1^*, y_2^*, y_3^*\}$, but again the equilibrium property is preserved. Interestingly, we have $y_3^* < y_1^* < y_2^*$, which is not reflected in the noise conditions $\sigma_1 < \sigma_2 < \sigma_3$. This implies that the interaction between sensors is complex, which may not be appreciated by a more naive approach.

6.5.2 Multiple Utilities, Mixed Congestion

The previous utility function represents a limiting case, since $\int_0^1 f(\alpha) d\alpha \approx f(1)$. If the amount of congestion is increased even slightly, e.g. by lowering $c^{(i)}$ or by increasing the energy cost $G_e(\alpha)$, the threshold strategy $\mu(y^*)$ falls out of equilibrium in low noise conditions.

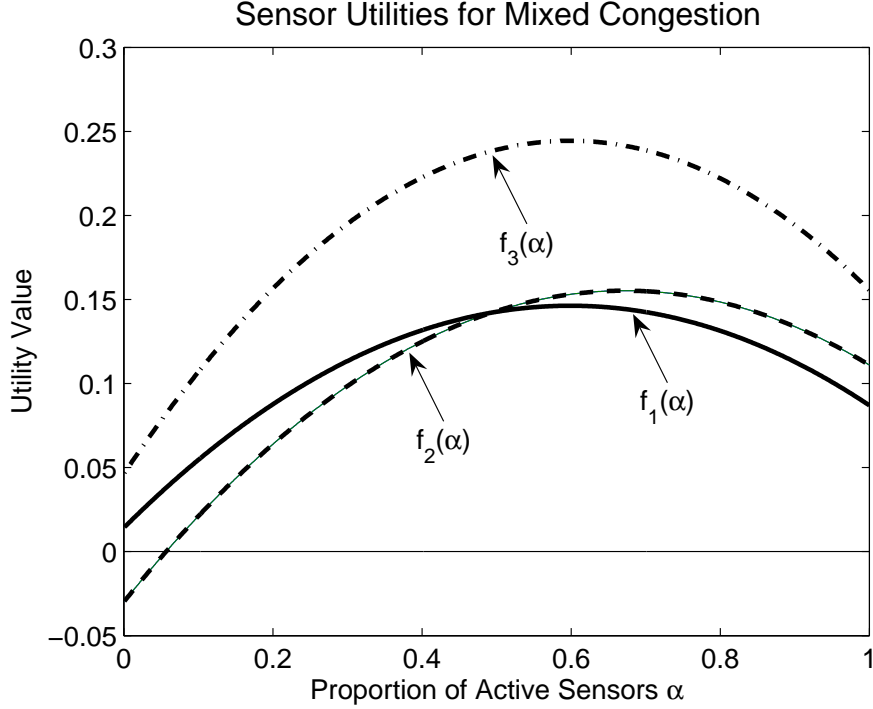


Figure 6.5: Sensor utility functions for second example. The degree of concavity in $f_J(\alpha)$, for classes $J = 1, 2, 3$, corresponds to each class's sensitivity to congestion; for sufficient concavity and low noise the threshold strategy fails to constitute a Nash equilibrium.

We define three sensor classes, as follows:

$$\begin{aligned}
 r_1 = 0.4 \quad c_1 = 0.04 \quad f_1(\alpha) &= -0.3685\alpha^2 + 0.4414\alpha + 0.0141 \\
 r_2 = 0.3 \quad c_2 = 0.05 \quad f_2(\alpha) &= -0.4106\alpha^2 + 0.5517\alpha + 0.0301 \\
 r_3 = 0.3 \quad c_3 = 0.06 \quad f_3(\alpha) &= -0.5527\alpha^2 + 0.662\alpha + 0.0462
 \end{aligned} \tag{6.45}$$

The first two classes have G_s and G_e as before, the third class finds it more costly to transmit when α is high. Since Type 3 has a higher contribution c_3 , we expect that it should have a lower threshold, but its higher cost of transmission suggests that the threshold will be optimal less often. Utilities for all three classes are shown in Figure 6.5.

For this scenario, we can compute the thresholds for small signals via (6.25) to yield $y_1^* = -2.8889$, $y_2^* = -2.7822$, $y_3^* = -2.8893$. However, Theorem 6.4.2 does not hold in this

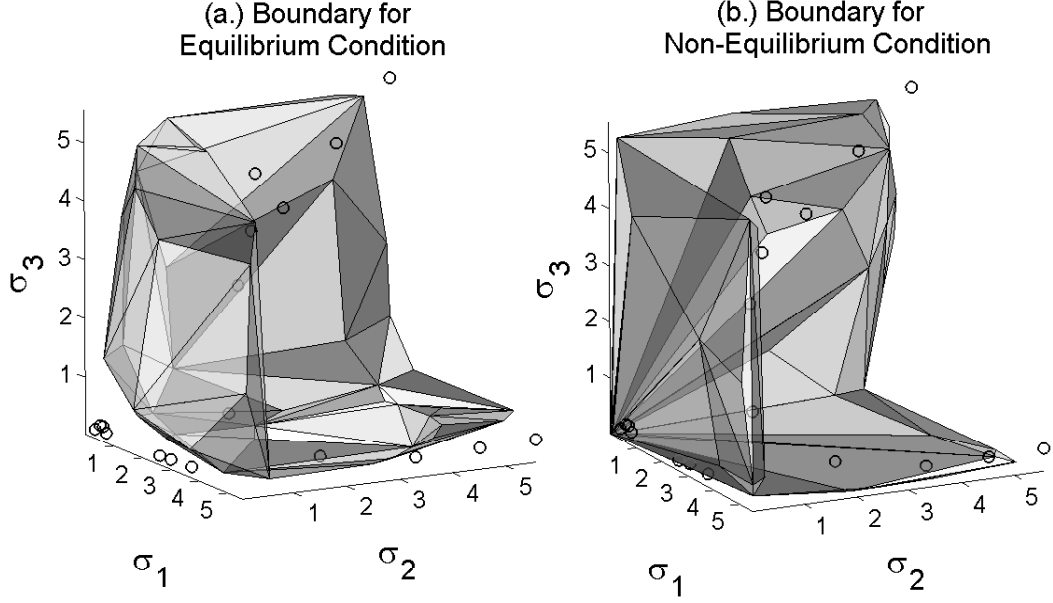


Figure 6.6: Illustration of the conditions for equilibrium in Theorem 6.3.2 for the second example. If the noise value $(\sigma_1, \sigma_2, \sigma_3)$ lies above the surface in Figure 6.6a, Theorem 6.3.2 guarantees the threshold strategy to be a Nash equilibrium. If it lies below the surface in Figure 6.6b, it cannot be a Nash. The circles represent test points which lie on the actual equilibrium boundary, between the two surfaces.

case, since, for example, if $y^{(i)} = -2.5$, then $y^{(i)} > y_1^*$ but

$$y^{(i)} < \frac{-1}{c_1} f_1 \left(\sum_{k \in \mathcal{I}} r_k I\{y^{(i)} > y_k^*\} \right) = f(1) = -2.175. \quad (6.46)$$

To establish for which noise values $\sigma_1, \sigma_2, \sigma_3$ equilibrium holds, we plot the bounds given in Theorem 6.3.2 in Figure 6.6 for a large number of test cases. The surface in Figure 6.6a corresponds to condition 6.26; if the noise triple lies *above* this surface, the threshold strategy is guaranteed to be a Nash equilibrium. Likewise, the surface in Figure 6.6b corresponds to condition 6.27; if the noise triple lies *below* this surface, the threshold strategy cannot be a Nash equilibrium. We also show a number of boundary points (shown as circles in Figure 6.6) at which equilibrium just holds.

For this scenario, the two conditions in Theorem 6.3.2 are close; the first surface lies just above the second. Moreover, the boundary points we computed lie just above the surface in Figure 6.6b. This indicates that (6.27) is a reliable condition for predicting Nash equilibrium of the threshold strategy for mode selection $\mu(y^*)$. This is fortunate since (6.27) is much easier to compute than (6.26); the function f'_J need be evaluated only at \mathcal{I} points, instead of on a continuum.

6.5.3 Convergence of Algorithm 6.4.1

The iterative procedure contained in Algorithm 6.4.1 for threshold computation need not always converge, since the map $T(y^*)$ may not be a contraction for all scenarios. In this section we test the algorithm on a large number of random cases to get a sense of when it will converge.

We again focus on the Gaussian case, since it is more interesting, and considered $N = 20,000$ scenarios with parameters randomly chosen according to a uniform distribution as follows:

- The prior distribution for X was Gaussian with mean m between -10 and 10 , and variance s^2 between 0 and 100 .
- Three classes $J = 1, 2, 3$ were considered, with utility: $f_J(\alpha) = a_J\alpha^3 + b_J\alpha^2 + c_J\alpha + d_J$, with each of a_J, b_J, c_J, d_J between -1.5 and 1.5 .
- The noise distribution for $W^{(i)}$ was Gaussian with zero mean and variance σ_J between 0 and 100 .
- The proportion of sensors of each class, r_1, r_2, r_3 was chosen uniformly.

A histogram showing the proportion of cases in which the threshold computation takes a given number of iterations to converge is shown in Figure 6.7. The figure shows the number of iterations drops rapidly, which is to be expected due to the exponential convergence of Banach's fixed point theorem. Such empirical results lend a high degree of confidence to the iterative procedure, even though theoretically, convergence depends on parameter selection. Note also that, although Algorithm 6.4.1 may produce a threshold, the equilibrium properties must still be checked.

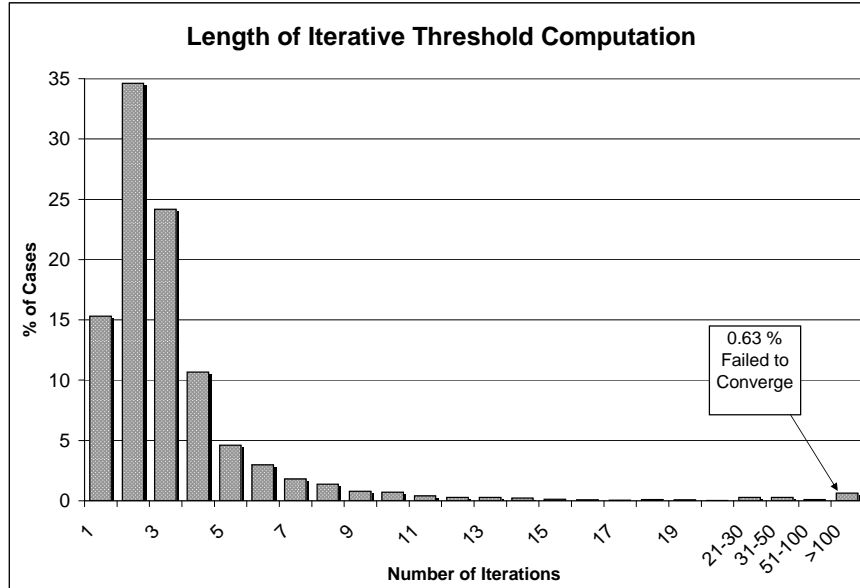


Figure 6.7: Number of iterations required for the threshold computation procedure of Algorithm 6.4.1 for 20,000 randomly selected scenarios. Only 0.63% of cases failed to converge within 100 iterations.

6.6 Conclusions

Large sensor populations and tight energy constraints have created a need for extremely light coordination protocols in many wireless sensor network applications; sensors must operate with a high degree of independence, based only on noisy information which varies from sensor to sensor. The theory of global games is a natural tool for providing insight into such cases as it considers sensors which act in isolation, with imperfect information and without complete knowledge of the information acquired by others. The application of global games developed here provides fresh perspective on the problem of decentralized, self-configuring sensor network design, allowing sensors to act independently while accounting for such issues as bias, algorithm complexity, and noise.

In general, it is impossible for sensors acting under these constraints to perform perfectly, since they must predict instead of react to their counterparts' actions, based on imperfect information. However, our analysis reveals that there are conditions under which self-configuration and decentralized operation are straightforward, leading to performance guarantees in the form of a Nash equilibrium for simple threshold behaviour. Moreover, by providing the proper motivation to each sensor, this Nash equilibrium can be made to reflect

the overall design goals of the sensor network. The result is a truly decentralized solution, in which sensors use local information to independently contribute to a global goal.

Our results indicate that a straightforward, threshold Nash equilibrium for mode selection exists when congestion is low or noise is high, and we give several simple tests for checking for such equilibria, especially in the limiting, low-noise case resulting from aggregated information. This extends the known existence of Nash equilibrium threshold policies from supermodular global games to global games with congestion and multiple player classes. The ability to relax the supermodularity conditions for a global game suggests there may be a weaker general condition for the existence of threshold strategies, much as the single-crossing condition is a weakening of supermodularity for traditional game theory. In addition, we show in Section 6.4.2 that, even when an equilibrium threshold does not exist, it may still provide near equilibrium behaviour and thus represents a viable solution. These findings point to the general applicability of global game theory to sensor network problems.

Bibliography

- [178] P. Biswas and S. Phoha. Self-organizing sensor networks for integrated target surveillance. *IEEE Transactions on Computers*, 55(8):1033–1047, 2006.
- [179] H. Carlsson and E. van Damme. Global games and equilibrium selection. *Econometrica*, 61(5):989–1018, 1993.
- [180] C. Chamley. *Rational herds: economic models of social learning*. Cambridge, 2004.
- [181] L. Clare, G. Pottie, and J.R. Agre. Self-organizing distributed sensor networks. *Proceedings of SPIE*, pages 229–237, 1999.
- [182] A. Aroraa et. al. A line in the sand: a wireless sensor network for target detection, classification, and tracking. *Computer Networks*, 46(5):605–634, 2004.
- [183] A.J. Goldsmith and S.B. Wicker. Design challenges for energy-constrained ad hoc wireless networks. *IEEE Wireless Comm.*, 9(4):8–27, 2002.
- [184] R. Guesnerie. An exploration of the eductive justification of the rational expectations hypothesis. *American Economic Review*, 82:1254–1278, 1992.
- [185] J. Hartigan. *Bayes Theory*. Springer-Verlag, 1983.
- [186] Kyle Jamieson, Hari Balakrishnan, and Y.C. Tay. Sift: a MAC protocol for event-driven wireless sensor networks. In *Third European Workshop on Wireless Sensor Networks (EWSN)*, Zurich, Switzerland, February 2006.
- [187] L. Karp, I. Lee, and R. Mason. A global game with strategic substitutes and complements. *Department of Agricultural and Resource Economics, UCB. CUDARE Working Paper 940.*, 2004.
- [188] W. Kirk and B. Sims. *Handbook of metric fixed point theory*. Kluwer Academic, 2001.
- [189] J. Liu, J. Reich, and F. Zhao. Collaborative in-network processing for target tracking. *Journal on Applied Signal Processing*, pages 378–391, 2003.

- [190] A.B. MacKenzie and S.B. Wicker. Game theory and the design of self-configuring, adaptive wireless networks. *IEEE Communications Magazine*, pages 126–131, November 2001.
- [191] M. Maskery and V. Krishnamurthy. Decentralized activation in a zigbee-enabled unattended ground sensor network: a correlated equilibrium game theoretic analysis. *submitted to IEEE Transactions on Signal Processing*, 2006.
- [192] D. McErlean and S. Narayanan. Distributed detection and tracking in sensor networks. *Asilomar Conference on Signals, Systems and Computers*, pages 1174–1178, 2002.
- [193] J. Mirkovic, G. Venkataramani, S. Lu, and L. Zhang. A self-organizing approach to data forwarding in large scale sensor networks. *Proceedings of IEEE ICC*, pages 1357–1361, 2001.
- [194] S. Morris and H. S. Shin. Global games: Theory and applications. Cowles foundation discussion papers, Cowles Foundation, Yale University, September 2000.

Chapter 7

Conclusion

The previous chapters have discussed, at length, several game theoretic approaches to design and analysis related to decentralized systems of wireless-enabled devices. This chapter completes the study by summarizing the approaches of Chapters 2 to 6. Common features of the game theoretic approaches are highlighted in Section 7.1, and the major results and lessons are reviewed in Section 7.2. A discussion of potential future work is given in Section 7.3.

7.1 Common Approach to Game Theoretic Design and Analysis

The game theoretic approaches to design and analysis of large-scale, decentralized wireless-enabled systems discussed in the preceding chapters all have several features in common. These features stem from the general commonality that each system considered comprises of physically separate components which are nevertheless connected in several ways. To provide a deeper understanding, the commonalities of the game theoretic design and analysis process are highlighted below.

7.1.1 System Objectives and Capabilities

The first step, assessing overall system objectives and taking stock of available resources, is standard in almost all engineering design or analysis problems, decentralized or not, and the approach here is no different. A mathematical model is developed which adequately represents the fundamental problems to be solved, providing a logical and sound basis for proposed solutions. The components of our models for decentralized missile deflection, monitoring and communication systems are briefly discussed below.

Fundamental Considerations: The fundamental decentralized systems models developed here are similar to those encountered in many *centralized* engineering/decision prob-

lems, consisting of a global objective, specification of capabilities, and a description of the operating environment, as listed below:

- **Global Objective:** This is a statement of the overall system goal, phrased as maximizing a utility function, which maps given environmental conditions and actions to the real line. The objective in Chapters 2 and 3 were to maximize the minimum distance between task group and missile, or to minimize the probability of a missile hit, encoded into a utility function for the stochastic game. Chapters 4 and 6 presented a global objective of maximizing overall information gathered by a sensor network while minimizing energy usage among individual nodes. Chapter 5 considered the objective of maximizing the minimum satisfaction level of a number of competing radio spectrum users.

The main consideration is to accurately reflect the desired outcome of the problem, incorporating trade-offs at all levels. These include *internal* trade-offs, such as between a sensor's contribution to monitoring and its energy consumption, *external* trade-offs, such as fairness between cognitive radio users or Naval platforms, and *global* trade-offs, such as balancing an end-user's quality of sensor information vs. the expected functional lifetime of the sensor network.

- **Capabilities:** A definition of the action set of the system as a whole. Since the system is comprised of multiple components, this is defined as an appropriate vector in a well-defined action space. The joint action spaces considered here were the set of missile countermeasure actions in Chapters 2 and 3, sensor mode selection in Chapters 4 and 6, and radio channel selection in Chapter 5. Constraints on actions, such as the constraints on countermeasure deployment considered in Chapter 2, are also accounted for here.
- **Environment:** This is what the system interacts with in pursuit of its objectives. These were the dynamic missile threat in Chapters 2 and 3, moving targets and energy reserves in Chapters 4 and 6, and primary radio users in Chapter 5. An appropriate state space and any laws for state dynamics is also specified. This may reflect controlled Markovian dynamics as in Chapters 2 and 3, or slowly varying (or slow exogenous Markovian) dynamics as in Chapters 4, 5, and 6.

Relationship to Global Solution: The mathematical model obtained through the above process should be sufficient (in theory) to determine a global solution to the problem in question, via methods such as Markov decision processes, hidden Markov models,

constrained static optimization, etc. It should also include some indication of the types of information and objectives that may naturally be assumed by each player in a game theoretic version of the same problem. To proceed with a game theoretic treatment of the same problem, a reformulation must take place. That is, the problem is posed from the point of view of multiple, interdependent decision makers. Solution of the game then requires simultaneous solution of these separate problems. This solution can be obtained either before implementation, as in Chapters 2 and 3, or during implementation, as in Chapters 4, 5, and 6. The nature of the environment, objectives and computational resources available should determine which method is most appropriate.

There are several benefits to beginning with a global version of the problem in question. First, it involves the formulation of global objectives, which can be used as a guide for creating local incentives (below). Second, solving the global problem first can provide a motivation for game theory, by showing where a centralized approach may be infeasible due to complexity, communication limitations, etc. Similarly, if the centralized approach is feasible, we may save considerable time by implementing it instead. Third, the global optimum provides an analytical point of comparison for the game theoretic equilibrium, so that one can judge any performance degradation between the centralized and decentralized versions of the problem, and trade this off with the extra overhead required for centralization.

7.1.2 Component Awareness

Often, the same conditions which drive the need for a decentralized systems also make it difficult for individual components to attain complete awareness of their environment. That is, if it is not feasible for a central authority to control components, it is also likely that information about the environment will also be difficult to propagate through the system. However, even local awareness can be sufficient for a game theoretic approach, a distinct advantage over more wholistic tools. For example, in Chapter 4 sensors could only detect local targets, but a game formulation was still possible by allowing sensors to use their history of MAC successes/failures to obtain a maximum likelihood estimate of the number of nearby active sensors. A knowledge of the type of component awareness is important for game theoretic analysis since it defines the scope of the interaction between components, which may then be related to the system's overall performance. Some of the considerations involved in component awareness are discussed below.

Requirement for Interaction: The awareness level of individual components is also important for interaction between individuals in the game. A critical requirement is that

each decision maker in the game should be sufficiently aware of its surroundings that, there exists at least one decision maker whose actions have a direct effect on the its observations, and vice-versa. Interaction can usually be described using a network, or graph, with decision makers as nodes and directed edges from i to j if the actions of decision maker i affect the observable environment of j . The natural requirement then is that the graph be *connected*, so that the action of any decision maker i will, by a series of observations and adjustments by intermediate players, affect the outcome of any other decision maker j .

Generating Awareness: This thesis has analyzed several possible mechanisms by which the awareness necessary for interaction may be generated. In Chapters 2 and 3, environmental (missile track) awareness was common knowledge to all players, either shared among players or assumed to be common by reasoning that each player could generate their own high-quality environmental awareness. On the other hand, awareness of other players' actions in the first chapters was completely restricted; players could not observe each others' present or past actions, and were therefore required to compute the equilibrium strategy in advance.

In Chapters 4 and 5, the *effects* of past actions were observable, through model-based maximum likelihood estimation based on analysis of MAC layer performance of the players' transmission efforts. In Chapter 4, the environmental awareness of sensors was localized, and the mutual awareness of sensor activity was localized in the sense that only the activity of sensors in the same cluster, or in neighbouring clusters, was observable. Moreover, mutual awareness in both chapters was based on the players' own abilities to gather information using sensors, frequency scanners, etc. and is therefore incomplete (based on averaging and statistics) and subject to noise.

Chapter 6 considered that each player was capable of *noisy* measurements of a global value correlating all players' utilities, upon which predictions were made and actions taken. In the decision problem itself, players did not have to be aware of each others' actions, as long as they could predict how others would behave. They did, however need to be aware of the number of other players of different types, so as to be capable of making such predictions. The uncertainty in measurements is explicitly taken into account, and actually improves performance by decoupling sensor actions so that action diversity can take place.

7.1.3 Local Incentives

One of the most challenging aspects in game theory is the specification or discovery of plausible local incentives or utility functions which would lead to satisfactory overall per-

formance approaching the global optimum, or which accurately model the system at hand. Due to limited component awareness, it is usually infeasible for each player to adopt the global utility function as a local incentive, and even when this is possible there may be coordination problems which prevent players from jointly implementing a globally optimal solution. The design of new incentives and consideration of existing ones is discussed below.

Incentive to Contribute: To deal with the issue of limited awareness, we proposed in Chapters 4 and 6 that sensors attempt to maximize their best estimate of their *contribution* to the global objective, instead of using the global objective itself as incentive. This translates into each sensor estimating the current slope of the global objective function, and acting to maximize a reward proportional to m , minus the local energy cost of its actions.

This technique has the benefit that each player will act to immediately increase the global objective. However, the resulting equilibrium behaviour has not been shown to be globally optimal, especially since players do not have the complete information necessary to fully determine whether their contribution is positive or negative.

Pricing and Incentive for Self-Interest: Another challenge occurs in the cognitive radio setting, where local incentives (maximize bandwidth) are already naturally in place. The challenge then is to convince the individual players to modify their incentives to align with global requirements, possibly through side payments or threat of sanction. In Chapter 4, this modification was done through a centralized pricing mechanism, intended to optimize global performance while still allowing most activity to be carried out in a decentralized manner. In Chapter 5, we analyzed how providing a few carefully chosen incentives to self-interested radio users could result in an improvement to system performance as a whole. Radios were allowed to seek out their required bandwidth, but were penalized if they either received more bandwidth than was required, or if they were involved in excessive collisions. These two incentives induce a more fair, efficient sharing of available resources, in accordance with design objectives.

This problem is also at the heart of the work on decentralized missile deflection in Chapter 3. In that chapter, some effort was required to encode the objective “defend asset set A from missile set M ” into a utility function. However, the equilibrium was complicated by the fact that each ship might have its own set of assets it wishes to defend. This might be due to political factors (alliances), personnel policy, or as a means of decentralizing a complicated objective.

Single Designer Requirement: There is nothing in the above discussion that prevents players, such as selfish cognitive radios, from defining their own utility to the detriment of other players. Indeed, in Section 5.6.1 it was shown that cognitive radios could benefit by disregarding their “concern” for the well-being of other users, but only at their expense. This apparent criticism points to an underlying assumption of the design and analysis results of this thesis, namely that a single designer is assumed to be ultimately responsible for setting incentives (via utility functions). If we allow human operators to manipulate the automated players’ utilities, then these operators become part of our system, and a higher designer (presumably economic) would be required to justify our approach, for example by introducing real financial penalties to players caught deviating from the prescribed game parameters.

7.1.4 Equilibrium Analysis

Even with players instilled with the proper incentives, it is difficult for each player to determine an optimal course of action. The challenge is essentially that each player must determine how to act, knowing that its environment (other players) will respond to any action in an intelligent fashion. This is much different from the traditional control systems which operate in an environment with predictable inputs and outputs. This section outlines some of the major considerations and approaches involved in the analysis of a game theoretic equilibrium.

Nash or Correlated Equilibrium: A central issue is the degree of player coordination. That is, whether a player has access to any extra signals from its “environment” (in particular indicating the intentions of other players), before choosing its own action. In our research this manifests as the difference between a correlated and a Nash equilibrium. We stated in the introduction that the set of correlated equilibria is a generalization in the sense that it contains the set of Nash equilibria. Another point of view is that a correlated equilibrium is a refinement, in the sense that it is the Nash equilibrium of a game augmented by communication. Although these seemingly conflicting interpretations present an interesting puzzle, the practical consequence is that communication in a game allows more flexibility of operation, and improved performance, while leaving intact the fundamental freedom of choice of its players.

In addition to the choice between Nash and correlated equilibrium. Two types of game theoretic analysis are possible, the *descriptive* approach, which analyzes equilibria in an effort to discern patterns of behaviour and analyze performance in a known decentralized

system, and the *prescriptive* approach, which analyzes how equilibria can obtain in a decentralized environment so as to operate the system in a “competitively optimal” fashion.

Prescriptive/Design Approach: The prescriptive approach is appropriate as a design tool for decentralized systems, either to discover rules of behaviour that should be designed into system components, as a fallback strategy when component coordination or cooperation is infeasible, or to provide a basis of comparison with other approaches. The work in this thesis can be construed to follow the prescriptive approach to equilibrium in several ways, listed below:

1. In Chapter 2 an iterative linear programming technique for calculating a Nash equilibrium in stationary strategies was presented. Players compute the equilibrium during a negotiation phase by following the iterative procedure, then execute their strategies independently. A method to deal with the “non-Markovian” objective of maximizing the minimum distance of approach of a missile was also prescribed, by augmenting the state space model, then showing that the structure of the augmented state space allows for efficient computation, by collapsing the linear programming optimization into a sequence of problems of lower complexity.
2. In Chapter 3 we presented a bilinear program whose solution is a stationary correlated equilibrium, and related this solution to the paradigm of Network Enabled Operations [197]. Several architectures are discussed for implementing the correlated equilibrium, by exchanging information through a central coordinator. Coordination, which is modelled as the joint observation of a common random variable, is critical to implementation of the correlated equilibrium. A nonlinear constraint on the mutual information inherent in this coordination mechanism was also proposed, for the case where communication between platforms is limited.
3. Chapters 4 and 5 contain an adaptive regret-based procedure for learning a correlated equilibrium strategy in a decentralized manner. This represents a method by which players can reach equilibrium by online methods, as opposed to precomputation. Further analysis shows that a suitably modified procedure is able to track the correlated equilibrium set as the underlying game changes, provided that the rate of adaptation is sufficiently fast compared to the rate of change of the game. The correlated equilibrium set is an indicator of the global performance of the system, providing a performance guarantee for these systems.

4. Chapter 6 analyzed a Nash equilibrium policy based on a single threshold, and particular focus was placed on when the threshold policy represented a Nash equilibrium. This provides a simple method of implementing an equilibrium in many cases, which is important for design of simple wireless sensor nodes.

Descriptive/Analytical Approach: In addition, descriptive elements appear in each chapter. These help measure the effectiveness of equilibrium concepts, thereby justifying their use, or provide a tool for understanding the role of component interaction in system behaviour. The major descriptive features are as follows:

1. In Chapter 2, we establish the existence of a Pareto optimal Nash equilibrium for the team game case. This shows that under cooperative conditions the Nash equilibrium of a stochastic game is indeed desirable from a design perspective.
2. Chapter 3 focuses on the communication requirements involved in implementing a stationary correlated equilibrium. An information theoretic measure is defined and analyzed in several examples to measure the expense involved in player coordination. The optimal correlated equilibrium was also described in terms of the military doctrine of Network Enabled Operations.
3. Chapters 4 and 5 analyze the correlated equilibrium set to determine its performance. Pareto optimality of some equilibria is established in Chapter 4, and the relationship between equilibrium and global performance is explored in both chapters.
4. In Chapter 6, the robustness of the threshold Nash equilibrium is examined, by establishing conditions on the utility functions and noise conditions under which this type of equilibrium holds. This allows for predictions of how long the threshold policy remains a Nash equilibrium when noise decreases, due to accumulation and recursive averaging of measurements by each individual sensor.

Practical Considerations: Ultimately, when a game theoretic approach is proposed for design or analysis, the equilibrium analysis should measure its feasibility. The qualitative benefits, such as simplicity, efficiency and robustness, should be weighed against the performance loss (or gain) of a game theoretic approach. Moreover, analysis should not only consider how well the equilibrium performs, but also how the equilibrium is reached in the first place. This will ensure that the game theoretic approach is not only reasonable, but also workable.

7.2 Evaluation of Results

As with any analytical, model-based approach, the results of the preceding chapters have their own benefits and drawbacks. In order to provide further insight into the game theoretic approach discussed above, this section provides an evaluation of these results, and consequently of their underlying game theoretic approaches.

7.2.1 Analytical Tools Used

Game theory provides a framework for considering a variety of engineering design and problems, but it remains only a framework; additional tools must be applied for complete analysis of the results implied therein. In particular, the equilibrium concepts central to game theory (which are similar to the concept of optimality in decision theory) achieve a balance in performance among conflicting interested parties, but to compute or learn this equilibrium requires several further analytical tools, discussed below.

Fixed Point Computation: To compute a Nash equilibrium, one must find the fixed point of a nonlinear system. This is not easy in general; known iterative procedures for computing Nash equilibria may not be attracted to a fixed point or even a limit cycle in all cases, but may instead be drawn to a strange attractor, essentially signalling failure of the procedure. On the other hand, some games possess certain structural properties that facilitate Nash equilibrium computation, such as potential games [208], and games with strategic complements and substitutes (supermodularity and complementarity) [213].

Structural Analysis: These structural properties are used in the computation Nash equilibria in Chapters 2 and 6, respectively. Chapter 2, which deals with stochastic games, uses tools such as dynamic programming and convex optimization [200, 202] to define a sequence of linear programming problems that converges to a Nash equilibrium. Because of the structure of the game, simple analytical tools (monotone convergence) yield the desired convergence results. In Chapter 6, expected utility theory allows us to “guess” a Nash equilibrium policy and check that it holds under nearly supermodular game conditions. This requires Bayesian reasoning techniques particular to “global game” problems [201, 204, 209].

Convex Optimization: Computation of a correlated equilibrium is simpler in a sense, as it only requires finding a randomized strategy that satisfies a set of constraints, which is easier than a fixed point problem. The convex optimization or feasibility problem [200]

characterizing the set of correlated equilibria [196] is specified in Chapters 3, 4 and 5. Chapter 3 concerns a multi-stage decision problem, so dynamic programming and nonlinear optimization are used to characterize the correlated equilibrium set. Otherwise correlated equilibria are simply represented by the solution to a linear feasibility problem.

Stochastic Approximation and Stability: Chapters 4 and 5 also study online adaptive filtering procedures for tracking the set of correlated equilibria of a slowly varying game. Convergence analysis tools from stochastic approximation theory [206, 212], such as differential inclusions [198] and Liapunov stability, are used to show that the procedures converge. Although these results are highly technical, the main results are formulated in terms of almost sure or weak (in probability) convergence [206]. Roughly speaking, the adaptive procedures need time to converge to equilibrium, and must retain a sufficiently large history of observations, but not so large that variations in the game are “averaged out.”

Other Tools: Other mathematical tools are used as the need arises. Maximum likelihood estimation is used in Chapters 4 and 5 to provide devices with mutual awareness, forming the basis for game interaction. Bayesian estimation is used in Chapter 6 to devise a predictor for the proportion of active sensors given a noisy, globally correlated signal. Stochastic optimization is proposed as a method for re-parameterizing device utility functions in order to improve design objectives in Chapter 4. Information theory is used in Chapter 3 to analyze the amount of communication required to implement a correlated equilibrium. Other potentially useful tools include graph theory for characterizing player interactions, and perturbation analysis, for analyzing the stability of a given equilibrium.

7.2.2 Strengths and Weaknesses

In Section 1.1.3, several benefits and drawbacks relevant to game theoretic design and analysis were listed. The basic idea, there and throughout this thesis, is that game theory offers simplicity and flexibility in exchange for limitations on global performance. This trade-off is echoed by a similar trade-off in experience: the game theoretic approach itself is relatively simple to implement, but it is difficult to say much about the global performance of the resulting system. Below we discuss some additional strengths and weaknesses of the game theoretic approach.

Equilibrium vs. Optimality: One potential weakness is that it is often difficult to reconcile the notion of equilibrium with traditional concepts of optimality. As often as possible, comparisons between equilibrium and optimality have been carried out in this thesis (see Section 7.1.4). However, to be meaningful, both the global utility and “player” utilities must be comparable, in the sense that they reflect some ultimate reality or design objective. Often, this is not the case: sensor utilities in Chapters 4 and 6 do not reflect true local objectives, but are merely convenient tools for reaching a global objective; cognitive radio global and player utilities are both real, but cannot be measured in common units. An exploration of these issues is given in [203]. Game theory is not meant to deal with centralized notions of optimality, but does succeed at providing excellent analysis of “competitive” optimality. This is extremely useful for analyzing systems that suffer from internal communication limitations, but may be inappropriate otherwise.

Randomization of Actions: An additional drawback to game theoretic design is the reliance on randomized strategies to maintain equilibrium, especially in Chapter 3. This may be perfectly acceptable in economic applications, where randomization is used to guard one’s strategy against competitors, but one may object to such “holding back” of information in an engineered game consisting of automated devices. Fortunately, randomization does have some justification in decentralized systems, such as to ensure fairness for random access communications techniques (such as ALOHA), and more generally is thought to be a cornerstone of complex adaptive systems [207], allowing random “experiments” as a means to efficiently explore unknown environments. However, it is difficult in many cases to say what level of randomization promotes fairness, and what level is the result of destructive, self-interested behaviour. A final justification for random actions is given in [195], which shows that randomization is a natural phenomenon of overly-constrained Markov decision process problems.

Computational Complexity: A major advantage of decentralized systems design is the elimination of computational burden required to centrally coordinate a large number of devices. Since the number of possible actions of the systems considered in this thesis grows exponentially with the number of devices, decentralization represents a huge reduction in complexity, bringing intractable problems into the realm of feasibility. However, while the number of options a single device in a decentralized system has to consider are relatively small, these devices are presented with an additional burden; they must now track and/or predict the actions of their counterparts in order to select an appropriate action in response.

The complexity trade-off is usually worthwhile; Chapters 2, 4, 5 all give relatively low-complexity solutions to what would be an intractable centralized problem. Chapter 3, on the other hand, falls short, proposing a nonlinear optimization to compute the correlated equilibrium, where a linear optimization on the same decision space could yield a global optimum. On the other hand, the correlated equilibrium property here is more general, as it represents a solution which simultaneously optimizes several conflicting objectives.

Precomputation of Solutions for Stochastic Games: Even if they may represent a simplification of the global problem, the computational complexity (discussed above) required for computing stochastic game equilibria in Chapters 2 and 3 are sufficiently high that solutions must be pre-computed in practical situations. This is particularly true for the force protection scenario, where real-time computation of countermeasure policies (through machine learning or optimization) would be impossible during the few minutes a missile attack is expected to last. This represents a drawback in that the missile deflection policies used must be drawn from *anticipated* scenarios, involving missile types that act in known ways. If a new missile type or unanticipated scenario is encountered, the lookup-table method of implementing pre-computed policies fails to be optimal or in equilibrium. In this case one is forced to search for the nearest appropriate policy.

Subjectivity of Utility and System Parameters: Game theory describes the outcome of a game consisting of players with given interactions and given objectives. As such it does not systematically tell us what objectives are appropriate, either in a global sense or in a player-specific sense. It is up to the practitioner of the game theoretic design or analysis to determine the overall objective, and to test the player objectives to determine how well the game outcome satisfies this. We have demonstrated several approaches for dealing with the fundamental subjectivity of utility choices, and of other parameters in the system, as follows:

- In Chapter 2 we conducted a sensitivity analysis of system performance vs. game parameters, in order to demonstrate that variation of the system parameters was not extremely detrimental to the performance of the Nash equilibrium missile deflection policy.
- In Chapter 3, we showed how a myopic “shortcut” to equilibrium computation provided similar smooth degradation in performance of the correlated equilibrium for missile deflection. This shows that performance is relatively insensitive to subjective

design choices on whether to fully compute or approximate equilibrium policies.

- In Chapter 4, we introduced a centralized, stochastic-approximation-based pricing method for manipulating the utility functions of individual sensors, to improve a given global performance metric. This pricing method, which need only be carried out on a slow time scale, is a straightforward answer to the problem of choosing player utilities to accomplish a known objective. That is, pricing provides a partial escape method for the subjectivity in choosing player utilities; the parameters need not be fixed, but can be adjusted in real time to fulfill a global objective.
- Chapter 6 shows how the degradation in equilibrium performance may not always be smooth. That is, some subjective decisions about system parameters lead to catastrophic changes. In particular, as sensors in a global game become more certain about their measurements, there may be a single bifurcation point at which the described simple threshold strategies no longer comprise a Nash equilibrium. At this point only highly complex strategies can comprise an equilibrium, and the feasibility of computing such strategies disappears abruptly.

In the real world, where rapid design and flexibility are often important considerations, game theory is often an appropriate tool for design and analysis. However, because it dispenses with traditional centralized organization, the answers obtained from a game theoretic analysis may lack some satisfactory qualities that have come to be expected from most engineering analysis. A full reconciliation of these ideas is a subject for future work.

7.2.3 Characteristics of Appropriate Applications

Game theory is practical for design and analysis when a centralized approach is, in some sense, not appropriate for the situation at hand. This, in turn, can usually be traced to a deficiency in the *information structure* of the system under consideration. Either components of the system cannot sufficiently communicate with each other, or they are designed not to, as a strategy for dealing with external uncertainty. These two key characteristics of applications that are well-suited to game theoretic analysis are discussed below.

Internally Limited Communication: Networked sensors for defence and monitoring and systems of cognitive radios are appropriate candidates for decentralized, game theoretic design and analysis because they exhibit *internally limited communication*. This justification was given in Chapters 2 and 3 as the consequence of possible malicious interference

with critical communications systems among Naval platforms, and in Chapters 4 and 6 as the consequence of wireless sensors being limited in both bandwidth and energy. Taken as a network of interconnected devices, components of these systems simply cannot feasibly communicate sufficiently to pursue a common objective in a coordinated fashion. Thus, a game theoretic approach is attractive since it allows for *self-coordination* of devices.

Internal Conflict: In the case of cognitive radio (Chapter 5) or alliances of *self-interested* Naval platforms (Chapter 3), the lack of a common objective is inherited from a higher decentralized system, that of human users. These users have evolved as components of a biological system whose objective is to perpetuate itself. The larger system navigates a complex environment by allowing individual components (humans) to adapt using local rules and information, much in the manner prescribed by the game theoretic design approach above. Game theory works within this framework by considering self-interested entities which carry out their higher objectives; it allows for automated devices that seek out and execute behaviours that are self-interested, whether or not coordination is possible.

Complex Adaptive Systems: This brings into light a parallel between decentralized technology and complex adaptive systems found in biology. These systems, such as social and disease networks, cellular metabolism, immunology, and ant colonies, all use a decentralized approach to create adaptive, global functionality [207]. The difference between the applications considered in this thesis and these more complex biological systems is as follows. We have, for the most part, tailored our applications to a specific, known environment. Biological systems, on the other hand, have arisen as a kind of parallel, stochastic optimization-based solution to functioning in an unknown environment. Individual components are simply experiments, and successful experiments get reinforced through interactive mechanisms. The game theoretic approach described in this thesis is not sufficient to fully deal with complex adaptive systems of this kind. However, it can provide a rich analysis of many of the interactions within these systems, which are an inspiration for engineers aspiring to create truly adaptive, self-regulating technologies.

Summary: Game theory is best applied to systems with multiple, dispersed decision makers possessing independent preferences, especially when complete integration of these decision makers would be too costly. The autonomy and diversity of preferences are a direct result of limited available information (such as in random experiments in uncertain environments), or of limited information flow (such as in bandwidth-limited scenarios).

Because game players have preferences over *joint* actions, there is also a sense in which information flows between decision makers. Therefore, at least some of the effects of any component's actions can be observed by at least some other components.

7.3 Future Work

Complexity-based theories, such as games and decentralized systems, are still in their infancy, and much work is left to be done in understanding, integrating, and utilizing the various concepts that have recently emerged as a result of work in this area. The payoffs that arise from a good understanding of these ideas are enormous, including a deeper understanding of social, physical, and biological systems, and the potential to create appropriate technology for interacting with these systems on their own terms. We conclude with a list of potential future work in this direction, along the lines of this thesis.

Extension of Applications: To maximize understanding and clarity, the scenarios in this thesis have been kept quite simple. However it is interesting to consider more complex scenarios for wireless sensor networks, cognitive radios, and other applications. Increasing the decision space or interaction of these devices will certainly lead to more complex behaviour. For example, sensors could be required to make packet routing and forwarding decisions, or to choose a particular sensing range or target type to monitor. Sensors or cognitive radios could be capable of message passing for partial coordination, or might even be required to participate in a market-based game for resource allocation. Extending the complexity of applications promises much richer behaviour, but computational considerations become paramount, and tools for simplifying the search for game solutions requires novel techniques such as graph theory [205], potential functions [210, 214], and sub/supermodularity [213].

Other applications, such as coordination of mobile agents (e.g. UAVs, MANETs) or competition between networked agents (e.g. internet-based auctions or trading agents for financial, information or energy markets) could also benefit from a game theoretic approach. The same cooperative or competitive approach used here for sensor networks and cognitive radios could be applied straightforwardly to these types of problems.

Convergence of the regret-based tracking approach in Chapters 4 and 5 has been established, but several improvements are still possible. The main result is that convergence to the *set* of correlated equilibria is guaranteed by following a simple adaptive procedure. However, one might ask whether it is possible to modify the procedure to guarantee con-

vergence to some desired subset. Early indications are that this is a difficult problem, as any obvious modification of the procedure destroys the underlying convergence properties established in [199]. Another possibility is to extend the analysis of the decreasing and constant stepsize stochastic approximation approach underlying this procedure to include an *adaptive* stepsize, which may exhibit improved performance by striking a compromise between short-term sensitivity and long-term performance.

Connection between optimality and equilibrium: Centralized and decentralized paradigms have repeatedly been contrasted throughout this thesis. However, it is necessary to properly compare, and eventually integrate, these two approaches. This requires either that the qualitative difference between concepts of group and individual optimality be reconciled, or that one is made subordinate to the other.

We have, for the most part, subordinated individual performance to group performance, proposing incentives for individual sensors or cognitive radios that contribute to a global design objective. However such an approach invariably leads to trade-offs, which manifest as player utility parameters to be optimized. Although we propose a centralized, stochastic-optimization based pricing strategy in Chapter 4 to partially address this question, this requires a central authority to intervene, albeit intermittently, somewhat defeating the game theoretic approach. Further study in economics is needed to see whether a decentralized, market-based pricing mechanism can be used to overcome these difficulties.

There also remains a fundamental design question that has only been addressed in an *ad hoc* manner in this thesis, namely what is the connection between component utility and overall system utility. A systematic analysis, showing the effect of changing component utility functions on global performance, given specified interaction rules, remains to be carried out.

Connection of decentralized approaches: Our last comment on future work is the integration of a broad range of decentralized techniques. Network and graph theory, multi-agent systems, game theory, economics, dynamic programming and complex adaptive systems all have striking similarities in that each requires the simultaneous solution to a system of coupled “optimizations.” Furthermore, recursive algorithms are the most popular and promising method for solving problems in these areas. A unified body of knowledge, relating these theories to each other (and to the material in this thesis) would be extremely useful.

There has already been considerable success in this area, for example graph theory and dynamic programming are closely related, and the theory of stochastic games [211] was

at the outset closely coupled with the concept of dynamic programming. Game theory itself has also been closely integrated with many aspects of economics, such as auctions and bargaining.

The connection with complex adaptive systems is somewhat more challenging, requiring a leap to interactive agents that experiment and adapt, carrying out exploration and exploitation in a broadly defined stochastic environment. The applications considered in this thesis represent only a first, tentative step toward a fully integrated theory that will one day allow researchers to design and deploy aware, flexible, self-regulating systems to enhance and augment all our endeavors.

Bibliography

- [195] E. Altman. *Constrained Markov decision processes*. Chapman and Hall/CRC, 1999.
- [196] R. Aumann. Correlated equilibrium as an expression of bayesian rationality. *Econometrica*, 55(1):1–18, 1987.
- [197] S. Babcock. Canadian network enabled operations initiatives. *Proceedings of the 9th International Command and Control Research and Technology Symposium*, 2004.
- [198] M. Benaïm, J. Hofbauer, and S. Sorin. Stochastic approximations and differential inclusions ii: Applications. Levine’s bibliography, UCLA Department of Economics, May 2005.
- [199] D. Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6:1–8, 1956.
- [200] S. Boyd. *Convex Optimization*. Cambridge University Press, 2004.
- [201] H. Carlsson and E. van Damme. Global games and equilibrium selection. *Econometrica*, 61(5):989–1018, 1993.
- [202] J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer-Verlag, 1997.
- [203] A. Haeberlen, A. Mislove, A. Post, and P. Druschel. Fallacies in evaluating decentralized systems. *Proceedings of IPTPS*, 2006.
- [204] L. Karp, I. Lee, and R. Mason. A global game with strategic substitutes and complements. *Department of Agricultural and Resource Economics, UCB. CUDARE Working Paper 940.*, 2004.
- [205] M. Kearns, M. Littman, and S. Singh. Graphical models for game theory. *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 253–260, 2001.
- [206] H.J. Kushner and G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer-Verlag, New York, NY, second edition, 2003.

- [207] M. Mitchell. Complex systems: network thinking. *Artificial Intelligence*, 170(18):1194–1212, 2006.
- [208] Dov Monderer and Lloyd S. Shapley. Potential games. *Games and Economic Behavior*, 14(1):124–143, 1996.
- [209] S. Morris and H. S. Shin. Global games: Theory and applications. Cowles foundation discussion papers, Cowles Foundation, Yale University, September 2000.
- [210] R. Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.
- [211] L. Shapley. Stochastic games. *Proceedings of the National Academy of Sciences USA*, 39(10), 1953.
- [212] J. Spall. *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. Wiley Press, 2003.
- [213] D. M. Topkis. *Supermodularity and Complementarity*, volume 2. Princeton University Press, Princeton, New Jersey, 1998.
- [214] M. Voorneveld, P. Borm, and F. van Megen. Congestion games and potentials reconsidered. Discussion Paper 98, Tilburg University, Center for Economic Research, 1999.

Appendix A

Unattended Ground Sensor Network: Capabilities and Objectives

To maximize energy efficiency in a UGSN, it is important to understand the capabilities and constraints imposed by sensor technology and the ZigBee protocol. In this section, we discuss the physical sensor characteristics, including the system topology, radio environment, energy resources, and capabilities and requirements of various sensor components.

A.1 Sensor Network Model and Architecture

We consider a network of L energy constrained sensor nodes labeled by $l = 1, 2, \dots, L$, deployed randomly in a plane. Nodes are capable of short-range wireless communication, and are equipped with vibration (e.g., acoustic) or electromagnetic (e.g., visual) sensor components. They collect and process target readings, make local decisions for activation control, and transmit data using the ZigBee/IEEE 802.15.4 protocol.

A ZigBee network consists of full function (FFDs) and reduced function devices (RFDs) (see [216] for an overview). FFDs communicate and route data to an end user in a multi-hop fashion, and coordinate and receive data from RFDs. RFDs only send data to nearby “master” FFDs. The resulting network is a *cluster tree*, with RFDs arranged in a star topology [218] around a web of FFDs. The FFDs manage the more complex and energy-intensive tasks of coordinating RFDs and routing data, allowing the latter to conserve energy.

Sensor nodes in the UGSN are assumed to be RFDs, which only transmit information and do not have any receiving capability beyond the rudimentary ability to sense when the channel is in use, and to detect acknowledgements, as described in Section A.2. This unidirectional communication feature greatly simplifies UGSN operation, and increases the lifetime of sensor nodes. FFDs are hubs, which receive, process and forward sensor data to

an end user. FFD functionality is assumed fixed, and we concern ourselves only with sensor node (RFD) behaviour; route optimization, topology control, and optimal transmission among hubs are beyond our scope. Each hub i specifies a locally unique channel for data reception, and a node is said to belong to Cluster i if it transmits on this channel.

Below we provide nominal specifications for various sensor node functions, including their power requirements and time scales. Exact power specifications are given in Table A.2, as derived from various sources, including the Crossbow Inc. Mica mote, and [215, 217].

1. *Sensing Components:* Each sensor node is equipped with passive detectors, which indicate the direction and intensity of signals emanating from a target (e.g. footfalls, motor vibrations), subject to measurement noise. An active sensor node may take measurements at rates of 200 kHz, potentially generating a large amount of data for transmission. The power requirement for target sensing is taken to be 10 mW, including sampling, amplifying and A/D conversion.

2. *Processor:* A dedicated processor handles sensor data, node activation and transmission control. If raw sensor measurements are taken at 200 kHz, and batches of measurements of size 100 are processed and encoded for transmission, then sensor information would be produced at a rate of 2 kHz. Activation control decisions must also be made to conserve battery power, at a relatively slow rate (e.g. 20 Hz) compared to sensing tasks. To achieve this, a typical processor consumes approximately 24 mW, as in the MICAz mote.

3. *RF Transceiver:* When active, the RF Transmitter consumes up to 31 mW of power, while the receiver consumes 35 mW. In our application, the transceiver uses the ZigBee/IEEE 802.15.4 protocol. We describe the medium access control layer for this protocol in Section A.2.

4. *Power Supply:* In [217], Lithium battery technology is specified to contain approximately 340 WH of power per 1000 cm³. Assuming a 8 cm³ battery pack, this yields 2.72 WH of power. From the above specifications, a completely active sensor node consumes up to 67 mW, while a sleeping node consumes as little as 1% of this. A sensor node is therefore capable of roughly 41 hours of active operation, or 169 days of standby operation.

A.2 ZigBee/IEEE 802.15.4 Communication Model

The ZigBee/IEEE 802.15.4 standard is currently a leading choice for low-power communication in wireless sensor networks. It employs a CSMA/CA scheme for multiple access data transmission, and is used in sensor nodes such as the MICAz mote. Two versions of the CSMA/CA scheme are specified, depending on whether the network is beacon-enabled

or not. We focus on the non-beacon-enabled (unslotted CSMA/CA) mode, which is more energy efficient since the sensor node receivers do not need to switch on periodically to synchronize to the beacon [219]. The scheme is described below, with default parameters as in Table A.1:

Algorithm A.2.1 Unslotted CSMA/CA: Divide time into 0.32 ms units (the minimum time needed to detect an idle channel and prepare to transmit, see [216]). A sensor node with data to send executes the following steps:

1. Initialize the *number of backoffs performed* $NB = 0$, the *contention window size* $CW = 0$, and the *backoff exponent* $BE = BE_{min}$.
2. Wait R time units, where R is drawn from a discrete uniform distribution on $[0, 2^{BE}]$.
3. Perform a *clear channel assessment* (CCA) by listening to the channel.
 - (a) If the channel is idle, transmit if $CW = 0$. Otherwise, decrement CW and return to Step (2.)
 - (b) If the channel is busy, increment NB and BE to a maximum of BE_{max} , and reset $CW = 0$.
 - i. If $NB \leq NB_{max}$, return to Step (2.)
 - ii. If $NB > NB_{max}$, report a channel access failure, and discards the packet.

Despite the backoff algorithm, there may still be channel collisions. For this reason, transmissions are acknowledged by the receiver. If a packet is not acknowledged within the required period, $T_{AF_{max}}$, up to three retransmissions will be attempted in subsequent frames. Note: for slotted CSMA/CA in IEEE 802.15.4, the contention window CW is initialized at 1.

The following model, due to [219] is used to analyze the performance of Algorithm A.2.1. Let q be the probability that a node is transmitting at a given time. Assuming uniform transmissions, this may be taken as $q = T_{TX}F$, where F represents the number of transmissions per second, and $T_{TX} = 8(D + H)/B$ is the packet transmission time. (D is the payload size in bytes, H and B are as in Table A.1.) Let η denote the number of active sensor nodes in a cluster. The probability that the channel is clear in Step (3.) is:

$$p(\eta) = (1 - q)^{\eta-1}, \quad (\text{A.1})$$

Table A.1: IEEE 802.15.4 Standard Values

Description	Variable	Value
Min. Backoff Exponent	BE_{min}	3
Max. Backoff Exponent	BE_{max}	5
Max. # Backoffs	NB_{max}	4
Packet Header Length	H	33 bytes
Transmission Rate	B	250 kbps
CCA Time	T_{CCA}	0.128 ms
Max. ACK Wait Time	$T_{AF_{max}}$	0.864 ms
Mean ACK Wait Time	T_{AF}	0.528 ms

This is used to compute the probability $s(\eta)$ of successful transmission, and the average number of backoffs $R(\eta)$ as:

$$s(\eta) = \sum_{k=0}^{NB_{max}} p(\eta)(1 - p(\eta))^k, \quad (\text{A.2})$$

$$R(\eta) = \sum_{k=0}^{NB_{max}} (k + 1)p(\eta)(1 - p(\eta))^k + (NB_{max} + 1)(1 - s(\eta)), \quad (\text{A.3})$$

where NB_{max} is as in Table A.1. The last term in (A.3) was not included in [219], but is critical in our study since s is non-negligible.

Define P_{TX} and P_{RX} to be the active transmitter and receiver power requirements, as in Table A.2 (we use the **Chipcon CC2420** transceiver chipset). Taken together with Table A.1, the average energy required per transmission attempt is:

$$\begin{aligned} E_{packet}(\eta) &= R(\eta)P_{RX}(T_{CCA} + T_{RX_{wu}}) \\ &\quad + s(\eta)(P_{TX}(T_{TX} + T_{TX_{wu}}) + P_{RX}(T_{RX_{wu}} + T_{AF})). \end{aligned} \quad (\text{A.4})$$

Functions $E_{packet}(\eta)$ and $s(\eta)$ are plotted for several values of F in Figures A.1 and A.2.

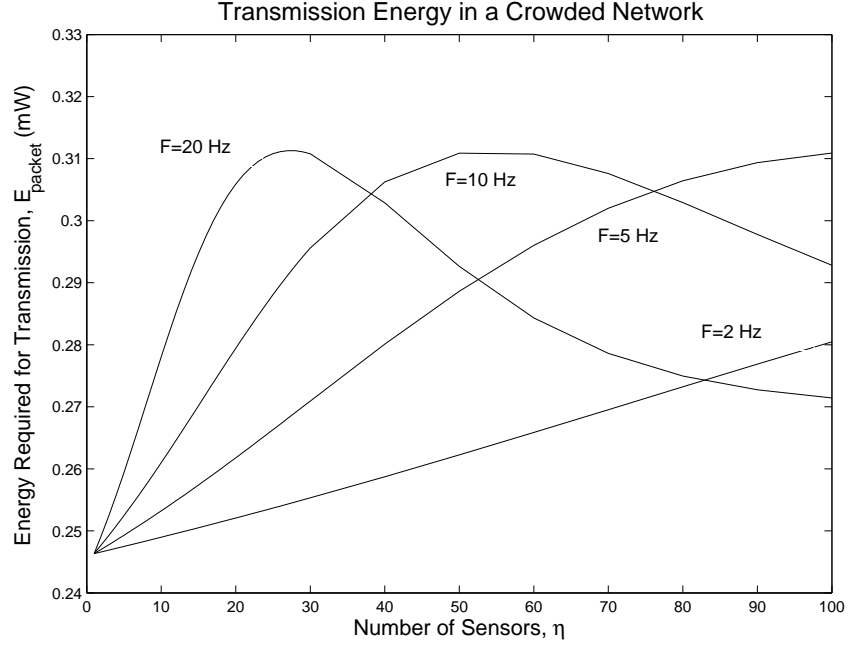


Figure A.1: Energy required for each packet transmission attempt in a network of η active sensor nodes, plotted for several values of F , the rate of transmissions per second per node. The energy decreases in crowded networks because the node often aborts transmission due to lack of a clear channel. Packets are of length 50 bytes, transmission power is 0 dBm.

Table A.2: Power Consumption and Warmup Times for Various Sensor Components

Device		Variable	Value
CPU	(Active)	P_{CPU}	24 mW
	(Sleep)	P_{CPU_0}	0.3 mW
Sensing	(Active)	P_S	10 mW
	(Sleep)	P_{S_0}	0.1 mW
Radio Rx	(Active)	P_{RX}	35.28 mW
	(Sleep)	P_{RX_0}	0.002 mW
Radio Tx	(Sleep)	P_{TX_0}	0.002 mW
	-25 dBm	P_{TX}	15.16 mW
	-15 dBm		17.48 mW
	-10 dBm		19.62 mW
	-5 dBm		22.09 mW
	0 dBm		30.67 mW
Tx Warmup Time		$T_{TX_{wu}}$	1.4 ms
Rx Warmup Time		$T_{RX_{wu}}$	1.4 ms

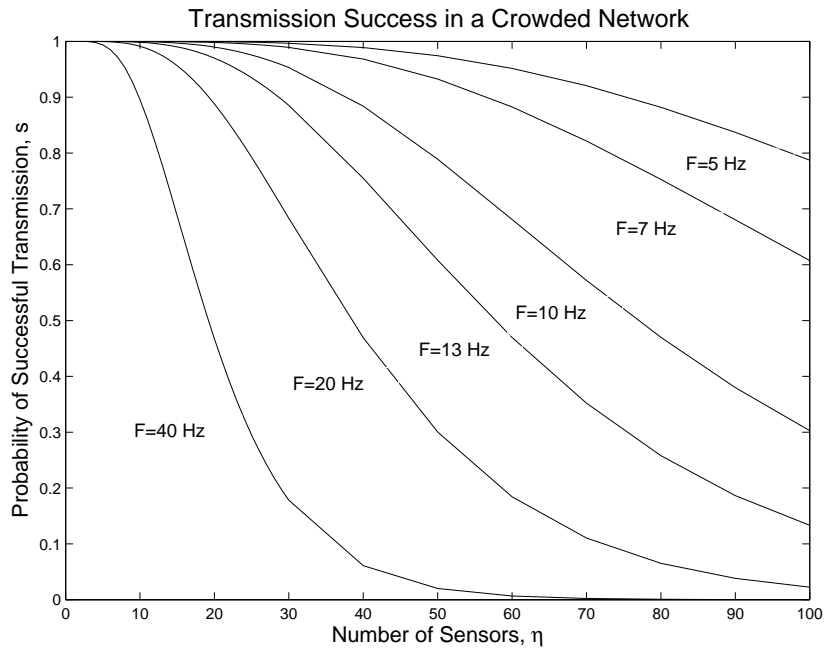


Figure A.2: Probability of successful packet transmission in a network of η active sensor nodes, plotted for several values of F , the rate of transmissions per second per node. Packets are of length 50 bytes, transmission power is 0 dBm.

Bibliography

- [215] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks*, 38(4):393–422, 2002.
- [216] E. Callaway, P. Gorday, L. Hester, J.A. Gutierrez, M. Naeve, B Heile, and V. Bahl. Home networking with ieee 802.15.4: a developing standard for low-rate wireless personal area networks. *IEEE Communications Magazine*, 40(8):70–77, 2002.
- [217] W.M. Merrill, F. Newberg, K. Sohrabi, W. Kaiser, and G. Pottie. Collaborative networking requirements for unattended ground sensor systems. *In Proc. IEEE Aerospace Conference*, 2003.
- [218] C.K. Singh and A. Kumar. Performance evaluation of an ieee 802.15.4 sensor network with a star topology. 2006. (preprint).
- [219] N. F. Timmons and W. G. Scanlon. Analysis of the performance of ieee 802.15.4 for medical sensor body area networking. *Proceedings of The First IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, pages 16–24, 2004.

Appendix B

Proofs of Results for Chapter 6

Proof of Lemma 6.2.1:

Strategy $\mu^{(i)}$ can be represented by the function:

$$q_{\mu^{(i)}}(Y^{(i)}) = \Pr(\text{“High_Res”} | Y^{(i)}). \quad (\text{B.1})$$

We then have

$$C^{(i)}(X, \alpha, \mu(Y^{(i)})) = h^{(i)}(X, \alpha) q_{\mu^{(i)}}(Y^{(i)}), \quad (\text{B.2})$$

where α is implicitly a function of μ . The objective (to be maximized) can be written as:

$$\begin{aligned} E[C^{(i)}(X, \alpha, \mu)] &= E[E[C^{(i)}(X, \alpha, \mu^{(i)}(Y^{(i)})) | Y^{(i)}]] \\ &= \int_{-\infty}^{\infty} E[h^{(i)}(X, \alpha) q_{\mu^{(i)}}(Y^{(i)}) | Y^{(i)} = y^{(i)}] p_{Y^{(i)}}(y^{(i)}) dy^{(i)} \\ &= \int_{-\infty}^{\infty} q_{\mu^{(i)}}(y^{(i)}) E[h^{(i)}(X, \alpha) | Y^{(i)} = y^{(i)}] p_{Y^{(i)}}(y^{(i)}) dy^{(i)}. \end{aligned} \quad (\text{B.3})$$

This suggests that an optimal strategy function $q_{\mu^{(i)}}(Y^{(i)})$ satisfies:

$$q_{\mu^{(i)}}(y^{(i)}) = I(E[h^{(i)}(X, \alpha) | Y^{(i)} = y^{(i)}] > 0), \quad (\text{B.4})$$

whenever $p_{Y^{(i)}}(y) > 0$, since this maximizes the expectation by removing the negative portions of the integral. \square

Proof of Theorem 6.3.1:

Part 1.) From Bayes' rule, it follows that:

$$p_{Y^{(i)}|X}(y|x, \varepsilon_J) = \frac{1}{2\varepsilon_J} I\{x - \varepsilon_J < y < x + \varepsilon_J\}, \quad (\text{B.5})$$

$$p_{X|Y^{(i)}}(x|y, \varepsilon_J) = \frac{1}{2\varepsilon_J} I\{y - \varepsilon_J < x < y + \varepsilon_J\}. \quad (\text{B.6})$$

The diffuse nature of the prior on X does not pose a problem, see [220]. Now, the expected reward (6.15) may be rewritten as:

$$\begin{aligned}
 & E[h_J(X, \alpha) | Y^{(i)} = y^{(i)}] \\
 &= c_J y^{(i)} + \frac{1}{2\varepsilon_J} \int_{y^{(i)} - \varepsilon_J}^{y^{(i)} + \varepsilon_J} f_J \left(\sum_{k \in \mathcal{I}} r_k \frac{1}{2\varepsilon_k} \int_{y_k^*}^{\infty} I\{x - \varepsilon_k < y < x + \varepsilon_k\} dy \right) dx \\
 &= c_J y^{(i)} + \frac{1}{2\varepsilon_J} \int_{y^{(i)} - \varepsilon_J}^{y^{(i)} + \varepsilon_J} f_J \left(\sum_{k \in \mathcal{I}} r_k \left[\frac{x - y_k^* + \varepsilon_k}{2\varepsilon_k} \right]_0^1 \right) dx, \tag{B.7}
 \end{aligned}$$

where $[x]_0^1$ denotes $\max\{0, \min\{1, x\}\}$. Substituting this into the general threshold form of (6.10) yields the result.

Part 2.) Having characterized the thresholds y^* , we consider conditions under which the threshold strategy $\mu(y^*)$ for mode selection is a Nash equilibrium. Recall (6.11), the condition for Nash equilibrium. Since the reward component f_J is bounded, (B.7) is negative for $y^{(i)}$ small enough and positive for $y^{(i)}$ large enough. Therefore, an equivalent condition for the threshold strategy to be a Nash equilibrium is that y_J^* is the unique zero of (B.7) for all $J \in \mathcal{I}$. That is,

$$E[h_J(X, \alpha) | Y^{(i)} = y^{(i)}] = 0 \Leftrightarrow y^{(i)} = y_J^*. \tag{B.8}$$

A sufficient condition for this to hold is that $dE[h_J(X, \alpha) | Y^{(i)} = y^{(i)}] / dy^{(i)} > 0$ everywhere, and a necessary condition is that this derivative is positive at y_J^* . Since:

$$\begin{aligned}
 \frac{d}{dy^{(i)}} E[h_J(X, \alpha) | Y^{(i)} = y^{(i)}] &= c_J + \frac{1}{2\varepsilon_J} f_J \left(\sum_{k \in \mathcal{I}} r_k \left[\frac{y^{(i)} + \varepsilon_J - y_k^* + \varepsilon_k}{2\varepsilon_k} \right]_0^1 \right) \\
 &\quad - \frac{1}{2\varepsilon_J} f_J \left(\sum_{k \in \mathcal{I}} r_k \left[\frac{y^{(i)} - \varepsilon_J - y_k^* + \varepsilon_k}{2\varepsilon_k} \right]_0^1 \right), \tag{B.9}
 \end{aligned}$$

the result follows. \square

Proof of Corollary 6.3.1:

This follows from Theorem 6.3.1 and the hypotheses $y_1^* < y_2^* < \dots$ and $y_J^* + \varepsilon_J \leq y_k^* - \varepsilon_k$

for all $J \neq k$ by considering a change of variables in (6.16):

$$\alpha(x) = \sum_{k \in \mathcal{I}} r_k \left[\frac{x - y_k^* + \varepsilon_k}{2\varepsilon_k} \right]_0^1 \quad (\text{B.10})$$

(A more complicated change of variables holds in the general case.) Over the domain of interest, we then have $d\alpha/dx = r_J/(2\varepsilon_J)$, and the limits of integration become $\alpha(y_J^* - \varepsilon_J) = \sum_{k < J} r_k$, and $\alpha(y_J^* + \varepsilon_J) = r_J + \sum_{k < J} r_k$. The result follows directly. \square

Proof of Theorem 6.3.2:

Part 1.) The expected reward (6.15) becomes:

$$\begin{aligned} E[h_J(X, \alpha) | Y^{(i)} = y^{(i)}] \\ &= c_J \nu_J^{(i)} + \frac{1}{\sqrt{2\pi}\omega_J} \int_{-\infty}^{\infty} f_J \left(\sum_{k \in \mathcal{I}} r_k Q \left(\frac{y_k^* - x}{\sigma_k} \right) \right) e^{-(x - \nu_J^{(i)})^2 / 2\omega_J^2} dx \\ &= c_J \nu_J^{(i)} + \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f_J \left(\sum_{k \in \mathcal{I}} r_k Q \left(\frac{y_k^* - \omega_J z - \nu_J^{(i)}}{\sigma_k} \right) \right) e^{-z^2/2} dz. \end{aligned} \quad (\text{B.11})$$

Setting (B.11) equal to zero and using the general threshold form of (6.10) establishes the result.

Part 2.) As in the uniform case, consider the derivative-based sufficient condition for equilibrium, $dE[h_J(X, \alpha) | Y^{(i)} = y^{(i)}] / dy^{(i)} > 0$, and the necessary condition that this derivative be positive at y_J^* . Differentiation of (B.11), is achieved by bringing the derivative past integral, noting that:

$$Q(h(y, z)) = \int_{h(y, z)}^{\infty} e^{-t^2/2} dt = - \int_{-\infty}^{h(y, z)} e^{-t^2/2} dt,$$

and that:

$$d/dy Q(h(y, z)) = -e^{-(h(y, z))^2/2} dh(y, z)/dy.$$

The result follows. \square

Proof of Theorem 6.4.2:

The proof of this theorem is broken into four parts, as follows:

a.) Beginning with the uniform case, recall (B.7):

$$\begin{aligned} E[h_J(X, \alpha) | Y^{(i)} = y^{(i)}] \\ = c_J y^{(i)} + \frac{1}{2\varepsilon_J} \int_{y^{(i)} - \varepsilon_J}^{y^{(i)} + \varepsilon_J} f_J \left(\sum_{k \in \mathcal{I}} r_k \left[\frac{x - y_k^* + \varepsilon_k}{2\varepsilon_k} \right]_0^1 \right) dx, \end{aligned}$$

where $[x]_0^1$ denotes $\max\{0, \min\{1, x\}\}$. Suppose that $|y^{(i)} - y_k^*| > \varepsilon_J + \varepsilon_k$ for all $k \in \mathcal{I}$. Then either $y^{(i)} - \varepsilon_J > y_k^* + \varepsilon_k$ or $y^{(i)} + \varepsilon_J < y_k^* - \varepsilon_k$. In the former case, it follows that $x > y_k^* + \varepsilon_k$ and the component of the integrand $r_k[(x - y_k^* + \varepsilon_k)/2\varepsilon_k]_0^1 = 1$. In the latter case, $x < y_k^* - \varepsilon_k$ and $r_k[(x - y_k^* + \varepsilon_k)/2\varepsilon_k]_0^1 = 0$. Combining these results over all $k \in \mathcal{I}$ yields a constant integrand $f_J(\sum_{k \in \mathcal{I}} r_k I\{y^{(i)} > y_k^*\})$ and it follows that:

$$E[h_J(X, \alpha) | Y^{(i)} = y^{(i)}] = c_J y^{(i)} + f_J \left(\sum_{k \in \mathcal{I}} r_k I\{y^{(i)} > y_k^*\} \right). \quad (\text{B.12})$$

b.) For the Gaussian case, recall (B.11) and suppose that $|y^{(i)} - y_k^*| > 3(\sigma_J + \sigma_k)$ and $\sigma_J \ll s$. Let $d_J = s/\sqrt{s^2 + \sigma_J^2}$ and $e_J = \sigma_J/(s^2 + \sigma_J^2)$. For $\sigma_J \ll s$, $d_J \approx 1$ and $e_J \approx 0$. Thus,

$$\begin{aligned} E[h_J(X, \alpha) | Y^{(i)} = y^{(i)}] \\ = c_J \nu_J^{(i)} + \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f_J \left(\sum_{k \in \mathcal{I}} r_k Q \left(\frac{y_k^* - d_J^2 y^{(i)} - \sigma_J(d_J z - e_J m)}{\sigma_k} \right) \right) e^{-z^2/2} dz. \\ = c_J \nu_J^{(i)} + \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f_J \left(\sum_{k \in \mathcal{I}} r_k Q \left(\frac{y_k^* - y^{(i)} - \sigma_J z}{\sigma_k} \right) \right) e^{-z^2/2} dz. \end{aligned}$$

Now, since f_J is bounded, the integrand is non-negligible when $|z| < 3$ (say). But by hypothesis,

$$\frac{|y^{(i)} - y_k^*| - 3\sigma_J}{\sigma_k} > 3, \quad (\text{B.13})$$

which implies that Q is either (approximately) zero or one over such z . The integrand is again a constant (sum of indicator functions) and (as before):

$$E[h_J(X, \alpha) | Y^{(i)} = y^{(i)}] \approx c_J \nu_J(y^{(i)}) + f_J \left(\sum_{k \in \mathcal{I}} r_k I\{y^{(i)} > y_k^*\} \right). \quad (\text{B.14})$$

- c.) Equations (B.12) and (B.14) imply that if noise is low, then “most” values of the expected reward can be calculated by simply assuming that all sensors of each type are either on or off. (We take $\sigma_J \ll s$ to simplify the presentation; a more precise version may be derived without this condition.) This is seen more easily by taking limits as $\varepsilon_J \rightarrow 0$ for all $J \in \mathcal{I}$ in the uniform noise case, or as $\sigma_J^2 \rightarrow 0$ for all $J \in \mathcal{I}$ such that $\sigma_J^2/\sigma_k^2 < \infty$ for all $J, k \in \mathcal{I}$ in the gaussian noise case, yielding:

$$E[h_J(X, \alpha) | Y^{(i)} = y^{(i)}] \rightarrow c_J y^{(i)} + f_J \left(\sum_{k \in \mathcal{I}} r_k I\{y^{(i)} > y_k^*\} \right), \quad (\text{B.15})$$

for all $y^{(i)} : y^{(i)} \neq y_k^*$ for any $k \in \mathcal{I}$.

- d.) Now, for $y^{(i)}$ outside the region specified in (6.36) or (6.37), the proper conditions hold. Combining the optimality condition (6.11) and (B.12) or (B.14) (noting that both are given by (B.15) in this case), we obtain the condition (6.35). If this optimality condition holds for all $J \in \mathcal{I}$ then by definition $\mu(y^*)$ is a Nash equilibrium for the specified $y^{(i)}$. \square

Bibliography

- [220] J. Hartigan. *Bayes Theory*. Springer-Verlag, 1983.