ASYMMETRIC INFORMATION GAMES AND CYBER SECURITY

A Dissertation
Presented to
The Academic Faculty

By

Malachi G. Jones

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
in
Electrical and Computer Engineering

School of Electrical and Computer Engineering
Georgia Institute of Technology
December 2013

ASYMMETRIC INFORMATION GAMES AND CYBER SECURITY

Approved by:

Dr. Jeff S. Shamma, Advisor
Professor, School of ECE
Georgia Institute of Technology

Dr. Eric Feron
Professor, School of AE
Georgia Institute of Technology

Dr. Faramarz Fekri
Professor, School of ECE
Georgia Institute of Technology

Dr. Doug Blough
Professor, School of ECE
Georgia Institute of Technology

Dr. Magnus Egerstedt
Professor, School of ECE
Georgia Institute of Technology

Date Approved: August 21, 2013

To Mom: You have loved me all my life

To Wanda: You are the love of my life

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Jeff Shamma, for his support and mentorship. Since I was the first PhD student to be recruited to his lab at Georgia Tech, I've had the honor and pleasure to see the lab flourish and grow. The first year in the lab, I felt like an only child that receives all the attention, nurturing, and time from their parent. I'll admit that I was a little jealous at first when new members of the lab began to arrive because now I had to share Jeff's time with them. After some time though, I began to warm up to my lab "siblings." *I've made friendships with my lab siblings that I'm sure will last a lifetime.* Jeff also did his best to ensure that we each had opportunities to spend quality time with him. Under his tutelage and mentorship, I have transformed from a kid fresh out of undergrad at UF to a mature and responsible researcher. Now as I take the next steps in life and leave the nest, I will always remember the lessons he has taught me about school, research, life, and now fatherhood.

Much thanks goes to the faculty members that have served on my PhD committee; the faculty members are Dr. Faramarz Fekri, Dr. Magnus Egerstedt, Dr. Eric Feron, and Dr. Doug Blough. Dr. Egerstedt and Dr. Feron were also on my Master's thesis committee and were very instrumental in the success of my thesis project. I had the opportunity to work closely with Dr. Egerstedt and his students to build the robotics lab from the ground up. Dr. Feron kindly donated a very expensive motion capture system to our robotics lab that enabled us to develop a GPS localization system.

Additionally, I would like to thank the professors who have inspired and encouraged me over the years. In particular, Dr. Herman Lam, Dr. Joachim Hammer, Peter Dobbins, and Dr. Marilyn Thomas-Houston at the University of Florida were each instrumental in my early development as an engineer and researcher. Dr. Gary May, Dr. Comas Haynes, and Dr. Raheem Beyah at Georgia Tech each provided insights and advice that influenced me both as a researcher and a person. I also would like to thank Professor Zhihua Qu at

UCF for inviting me to the summer NSF REU for robotics as an undergrad.

I've had the honor and privilege to work/collaborate with several outstanding people during my time at Georgia Tech. I'd like to first thank my Greek brothers. In particular, Georgios Chasparis was like an older brother to me in the lab. *Before becoming one of Jeff's students, I asked George about the PhD experience. He said that I would learn a lot about myself during the experience; that I did.* After G.C. graduated from the lab, Georgios Kotsalis took on the older brother role. He has also been a key collaborator for my cyber security work. Georgios Piliouras has also been very helpful and offering much appreciated advice about cyber security and the PhD process. I would be remiss if I didn't thank all my lab siblings; they are Michael Fox, Nicholas Dudebout, Yusun Lim, Ola Ayaso, and Yasin Yazicioglu. They have each been willing to listen to my ideas, provide suggestions/insights, and make the lab an enjoyable experience.

Finally, I would like to thank my family. They have offered their support and encouragement to me throughout the years as I have pursued my dreams. It is through their sacrifices and nurturing that I have had the privilege to be the first in my family to graduate from a university. Their continued support throughout the highs and lows of the PhD program will now culminate in me having the honor and privilege of being the first in my family to earn a Doctoral degree. *Thanks mom for all that you have done and the sacrifices that you have made as a single parent that have enabled me to become the person that I am today. I know dad would be proud if he was here.* Last, but not least, I would like to thank my wife Wanda. She is my best friend and has been by my side through the most challenging periods of my PhD experience.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

A cyber-security problem is a conflict-resolution scenario that typically consists of a security system and at least two decision makers (e.g. attacker and defender) that can each have competing objectives. For example, the objective of one decision maker (e.g. defender) could be to ensure that the security system operates at or above some threshold level of performance, while another decision maker's (e.g. attacker) objective could be to ensure that the system operates below this threshold. From the standpoint of a security researcher or practitioner, some questions of interest about the system are the following. *How will the security system perform over time? What will be the likely behavior of the key decision makers that control/influence the system? How does information asymmetry affect the behaviors of each decision maker?* Answers to these questions can lead to better system designs, better understanding of the system itself, and improved decision making ability by security professionals.

In this thesis, we are interested in cyber-security problems where one decision maker has superior or better information. Game theory is a well-established mathematical tool that can be used to analyze such problems and will be our tool of choice. In particular, we will formulate cyber-security problems as asymmetric information games, where game-theoretic methods can then be applied to the problems to derive optimal policies for each decision maker. *We will consider asymmetric games where the state of the world remains fixed over time (repeated games) as well as games where the state of the world can change from stage to stage according to a transition that is dependent on the current state and the moves of both players (stochastic games).* These optimal policies can then be used to predict the likely behavior of the decision makers and the performance of the system.

A severe limitation of considering optimal policies is that these policies are computationally prohibitive for repeated and stochastic asymmetric information games. In asymmetric information games, the computational complexity grows exponentially with respect

to the number of game stages. We address the computational limitations for repeated games and stochastic games in Chapters 3 and 4 respectively by considering suboptimal policies based on the ideas of model predictive control. A key property of the policies is that they remain computationally tractable as the number of game stages increase.

The contributions of this thesis are the following. For the repeated games, we introduce policy improvement methods for computing suboptimal policies that have tight performance bounds. We prove that the method's performance converges asymptotically to optimal with respect to the number of game stages. We also show that the improved policy can be computed by solving a linear program online whose complexity is constant with respect to the game length. Similarly, for the stochastic games, we derive bounds on the performance of the policy improvement methods and show that the policy can also be computed by solving a linear program online. We then demonstrate in Chapter 5 how the policy improvement methods can be applied to cyber-security problems to reduce the computational complexity of forecasting cyber attacks.

CHAPTER 1

INTRODUCTION

Society's critical infrastructure, which include its government, its military, and its businesses, rely on networked systems to function in a satisfactory manner. As a result of this reliance, networked systems have been a target of social groups that include criminals, foreign countries, and "hactivists[1]." Before these groups began to proliferate in the digital era and before networked systems became the lifeline of modern civilization, there were hackers whose objective was to break into computer systems for the challenge; there were no profit motives in it for them. As a consequence, companies and industries placed few if any resources into cyber security. In fact, one of the few areas where substantial resources were placed was cryptography.

Modern cryptography is a subset of cyber security that has since its infancy been mathematically grounded. A particular example is a public key encryption algorithm developed by three professors (Rivest, Shamir, and Aldeman) at MIT [1]. [2] The encryption algorithm, RSA, is based on number theory and relies on the premise that it is computationally infeasible to factor large prime numbers. Cryptography has progressed to a point that cyber hackers almost aways avoid attacking the algorithms directly [2]. Instead, hackers seek out alternative attack vectors that they are more likely to attack successfully. These alternative attack vectors consist of targeting aspects of the security system that are often based on heuristics instead of established mathematical theory. In light of the recent security threats that have impacted governments as well as businesses, an increasing emphasis has been placed on finding ways to mathematically ground other aspects of security outside of cryptography [3]. It is hoped that through this process security systems will be more secure and predictable.

[1] Anonymous is an example of a hactivist group whose goal is to attack cyber systems to make political statements.

[2] Diffie and Helman were the first to publish the public key encryption concept.

In this thesis, we consider developing mathematical models of security systems to analyze the system's performance and to predict the likely behavior of key decision makers that influence/control the system.[3] In particular, we are interested in the scenario where one decision maker (e.g. attacker) has superior information and seeks to actively conceal his knowledge when it is optimal to exploit the uncertainty of the other decision maker (e.g. defender). The uninformed decision must therefore use her current observations to improve her uncertainty so that she can make better decisions going forward into the future.

A mathematical model of a given cyber system can be used to formulate a security scenario into a strategic game with asymmetric information. Game-theoretic methods can then be used to derive optimal policies for each decision maker under zero-sum assumptions, and these policies can be used to predict the likely behavior of the decision makers and the performance of the system [4]. *Since the derivation of optimal policies is computationally prohibitive [5], the theoretical contribution of this thesis is to consider suboptimal policies that are based on the ideas of model predictive control. Specifically, we establish tight lower bounds on the performance of the suboptimal polices, and we show that these policies are computationally feasible*[4].

In the first half of the thesis we will discuss asymmetric information games. This discussion will include both repeated games and stochastic games, where the state of the world remains fixed in repeated games and the state can change during game play in stochastic games. In the second half we will discuss actionable cyber-attack forecasting. The objectives of actionable cyber-attack forecasting are to learn an attacker's behavioral model, to predict future attacks, and to select appropriate countermeasures to prevent future attacks. We will also demonstrate in the second half of the thesis how the computational results of the first half of the thesis can be used to reduce the complexity of cyber-attack forecast models.

[3] These decision makers are typically attackers and defenders.

[4] The computationally complexity of the optimal policies grows exponentially with respect to the number of stages of the game. In contrast, the complexity of the suboptimal policies remains constant.

1.1 Asymmetric Information Games

1.1.1 Repeated Games

In repeated zero-sum games, two players repeatedly play the same zero-sum game over several stages [6], [7], [8]. We assume that while both players can observe the actions of the other, only one player knows the actual game, which was randomly selected from a set of possible games according to a known distribution. The dilemma faced by the informed player is how to trade off the short-term reward versus long-term consequences for exploiting her private information, since exploitation can reveal the true state of the world [9]. Seminal work by Aumann and Maschler [10] derives a formulation for the value of the game, which quantifies the exploitation tradeoff, and also derives optimal policies for the informed player. Using this formulation to compute explicit optimal policies for games with multiple stages is computationally prohibitive.

We address the complexity issues of repeated games by deriving a suboptimal policy based on the concept of policy improvement. Policy improvement consists of first considering a policy whose performance is known or can be readily computed. *This initial policy is sometimes referred to as a baseline policy.* Next, a candidate policy is selected from some policy set. If the current candidate policy performs better than the baseline policy, the candidate policy becomes the new baseline policy. This knew baseline policy is an improvement over the previous baseline policy, hence a policy improvement. This process can continue in an iterative manner. If the policy set is finite, then an optimal policy with respect to the set can be found in a finite amount of time.

The baseline policy we consider for asymmetric games is a non-revealing policy, i.e., one that completes ignores superior information. An important characteristic of the non-revealing policy is that it is readily computable. The improved policy, which is implemented in a receding horizon manner, strategizes for the current stage while assuming a non-revealing policy for future stages. We derive bounds on the guaranteed performance of

the improved policy and establish that the bounds are tight. We then prove that the performance of the policy improvement methods converge asymptotically to optimal with respect to the number of game stages. Last, we show that the improved policy can be computed by solving a linear program whose complexity is constant with respect to the game length.

1.1.2 Stochastic Games

A stochastic game is a repeated game where the state can change from stage to stage according to a transition that is dependent on the current state and the moves of both players [11]. Of interest, is the scenario where one of the players has superior information in the stochastic game and also solely controls the state transitions. This scenario creates a complication for the informed player. The complication is that she must decide when and how to use her private information. *Note that by using her private information, she also reveals that information to player II, the uninformed player.* In their seminal work, Aumann and Maschler studied the special case, repeated games with asymmetric information, where every state is absorbing. This work provides insights into the issues that an informed player must address when playing against an opponent that can observe moves and use that information to better estimate the state of the world.

Although stochastic games with asymmetric information are more general than the repeated case, the ideas and formulations that Aumann and Maschler introduce extend to the stochastic case [9]. The computational challenges of computing an optimal strategy for the repeated case also extend to the more general stochastic case. We address the complexity issues of stochastic game in a similar manner to that of repeated games. Specifically, we consider applying approximate dynamic programming methods, which includes model predictive control, to yield computable suboptimal policies with guarantees

1.2 Cyber-Attack Forecast Modeling

The security community has placed a significant emphasis on developing tools and techniques to address known security issues. Some examples of this emphasis include security

tools such as anti-virus software and Intrusion Detection Systems (IDS). This reactive approach to security is effective against novice adversaries (e.g. script kiddies) because they typically use off-the-shelf tools and popular techniques to conduct their attacks [12]. In contrast, the innovative adversaries often devise novel attack vectors and methodologies that can render reactive measures inadequate. These pioneering adversaries have continually pushed the security frontier forward and motivate a need for proactive security approaches.

A proactive approach that we pursue in Chapter 5 is actionable cyber-attack forecasting. The objectives of actionable cyber-attack forecasting are to learn an attacker's behavioral model, to predict future attacks, and to select appropriate countermeasures [13]. The computational complexity of analyzing attacker models has been an impediment to the realization of reliable cyber-attack forecasting. We address this complexity issue by developing adversary models and corresponding complexity reduction techniques. We then introduce a heuristic for learning behavioral models of potentially deceptive adversaries online. Last, we consider a capture-the-flag problem, formulate the problem as a cyber-security game with asymmetric information, and demonstrate how the models and techniques developed in this chapter can be used to forecast a cyber-attack and recommend appropriate countermeasures.

1.3 Outline

This thesis is organized as follows. It is divided into three chapters, where the first two chapters cover repeated and stochastic games with asymmetric information, and the last chapter discusses cyber-attack forecast modeling. Each of these chapters is intended to be self-contained apart from some general mathematical preliminaries reviewed in the next chapter. Therefore a reader interested in only one of these three areas can pick up the necessary background in the next chapter and then skip to the chapter of interest. Readers

that have a solid understanding of asymmetric information games and dynamic programming can proceed directly to the chapter of interest. The last chapter summarizes the key concepts and ideas of applying the computational results of this thesis to cyber security games.

CHAPTER 2

MATHEMATICAL PRELIMINARIES

2.1 Game Theory

Game Theory provides methods to model decision problems as games, where each decision maker can have competing objectives; these methods can enable the interactions among decision makers to be studied. Although game theory has seen widespread applications in economics, it has also been used as a tool to study social and behavioral phenomena that include conflict resolution scenarios. Cyber security is a conflict resolution scenario that has recently garnered much attention from businesses and governments [14]. Since cyber security typically includes at least two decision makers (e.g. defender and attacker) with contradictory objectives who are competing with each other, game theory can provide a way to model this scenario as a strategic game to predict the likely behaviors/actions of the defenders and attackers.

A strategic game consists of a set of players $\{1, \ldots, P\}$, a set of actions A_i for each player i, and preferences over action profiles characterized by a function $u_i : A \mapsto \mathbb{R}$ [15], [16], [17]. An example of a conflict resolution scenario modeled as a game is the well-known *Prisoner's Dilemma*. In this scenario, there are two players, a set of actions $A_i = \{$*Cooperate, Defect*$\}$ for player i, and a utility function represented by the matrix

	C	D
C	$2, 2$	$0, 3$
D	$3, 0$	$1, 1$

(1)

Note that player I is the row player and player II is the column player. The best outcome for player I is if player II cooperates because player I can then achieve a payoff of 3 if she defects. This scenario (D, C) is an unlikely outcome because each player has a dominant strategy, which is to defect. *Note that a strategy is dominant if there exists no other strategy that can achieve a better payoff for all choices of the opponent.* Since defect is a dominant

strategy, a rational player will always select this action. Therefore, the likely outcome for the game is (D, D) with a payoff of 1 for each player. However, the best outcome for both players collectively is collaborate as (C, C) yields a payoff of 2. The dilemma is then if each player decides to cooperate in order to achieve the best payoff for the group, he/she risks the other player defecting and therefore receiving a payoff of 0.

2.1.1 Nash Equilibrium

The action profile a^* is a **pure Nash equilibrium** if for every player i

$$u_i(a^*) \geq u_i(a_i, a*_{-i})$$

for every $a_i \in A_i$, where a_{-i} denotes the collective actions of every decision maker except i (i.e. $a_{-i} = (a_1, a_2, a_{i-1}, \ldots, a_{i+1}, ...a_N)$). *Prisoner's Dilemma* has a pure Nash Equilibrium, which is the action profile (D, D). *Each player in a Nash equilibrium does not have a unilateral incentive to deviate from their current selected action.* However, it need not be the case that there is a unique Nash equilibrium or that an equilibrium exists at all. The following game, *Matching pennies* with payoff matrix

	H	T
H	$1, -1$	$-1, 1$
T	$-1, 1$	$1, -1$

(2)

is an example of a game that does not have a pure Nash equilibrium.

A more generalized concept of equilibrium in strategic games is a **mixed Nash Equilibrium**. *In his famous Ph.D. thesis [18], John Nash proved that every game with a finite number of players in which each player can choose from finitely many pure strategies has at least one mixed Nash equilibrium. Nash's work generalized the equilibrium concept of John Von Neumann. Von Neumann showed that a special class of games, zero-sum games, had at least one equilibrium [19].* A **mixed strategy** of a player in a strategic game is a probability distribution over the player's actions. Let α denote a mixed strategy profile.

The mixed strategy profile α^* is a mixed Nash equilibrium if for every player i

$$U_i(\alpha^*) \geq U_i(\alpha_i, \alpha^*_{-i}),$$

where $U_i(\alpha)$ is player i's expected payoff when all players randomize according to the mixed strategy profile α. In *matching pennies*, the mixed Nash Equilibrium is the following. Each player chooses an action of *Heads* or *Tails* with probability $\frac{1}{2}$. In general, a Nash equilibrium does not guarantee that the equilibrium strategies the players select are optimal. However, for zero-sum games, which is a subclass of games that includes *matching pennies*, the Nash equilibrium strategy is optimal.

2.1.2 Zero-sum Games

A **zero-sum game** is defined as a game where the payoff functions u_1 and u_2 of players I and II respectively are such that

$$u_1(a_1, a_2) + u_2(a_1, a_2) = 0$$

for every pair (a_1, a_2) of actions. In words, the players' interests in these games are diametrically opposed as what is the best outcome for one player is the worst outcome for the other. The payoff matrix for a zero-sum game has the form

	L	R
T	$A, -A$	$B, -B$
B	$C, -C$	$D, -D$

(3)

and can be represented in short-hand form as

	L	R
T	A	B ,
B	C	D

(4)

where player I's payoff is represented in equation (4) and player II's payoff is the inverse. Note that player I is the row player and maximizer and player II is the column player and minimizer.

2.1.2.1 Minimax theorem

The celebrated Minimax Theorem that was introduced by Von Neumann proved that every zero-sum game has a Nash Equilibrium and that the Nash equilibrium strategy is optimal for the two players. The formal theorem is presented below:

Theorem 1 ***(Minimax Theorem) [20]*** *For every two-person, zero-sum game, there exists a mixed strategy for each player, such that the expected payoff for both is the same value V when the players use these strategies. Furthermore, V is the best guarantee that each player can expect to receive from a play of the game; that is, these mixed strategies are the optimal strategies for the two players.*

The proof of the minimax theorem provides insights into how to solve zero-sum games by formulating the game as a linear program. To prove the minimax theorem, it is sufficient show that the following expressions

$$\text{payoff of row player} = \max_{x} \min_{y} x' M y \tag{5}$$

$$\text{payoff of column player} = \min_{y} \max_{x} x' M y \tag{6}$$

are equal in value, where M is the payoff matrix, x is the mixed strategy of the row player and y is the mixed strategy of the column player. It can be proved that equation (6) is the dual of equation (5). Therefore, by duality theory, the optimal objectives of the two LPs are the same.

2.1.2.2 LP formulation

The Nash Equilibrium for the zero-sum game can be computed by solving the following linear program [21]:

$$\begin{aligned} \min_{u} \quad & \sum_{i=1}^{|I|} u_i \\ \text{s.t.} \quad & Mu \geq 1' \\ & u \geq 0 \end{aligned} \tag{7}$$

Let $\beta = (\sum_{i=1}^{|I|} u_i^*)^{-1}$, then the value V is equal to β and the optimal mixed strategy x^* of player is $x^* = \beta u^*$.

2.1.3 Repeated Games

In a repeated game, a set of players play the same strategic game G over several stages. Each player can perfectly observe the moves of the other players and also keeps a history of the moves of the players for each stage of the game. *Strategies in these games are mappings from histories to actions.* The key difference between a one-shot game and a repeated game is that in a repeated game, each player has to take into account the impact of his/her current action on the future actions of the other players [22]. In other words, her previous actions can have a direct impact on her reputation and how the other players respond/retaliate. Repeated games can be divided into two subclasses, finitely and infinitely repeated games.

2.1.3.1 Finitely Repeated Games

A finitely repeated game is played over N stages. An example repeated game that will be considered is *Prisoner's Dilemma* [23]. To evaluate the Nash equilibrium of this game, it is useful to consider the last stage N. Observe that the actions of player i at this stage has no future consequence as the other players cannot punish player i because the game will have ended. Therefore, each player can view the last stage of the game as a one-shot game. In a one-shot game, each player will choose to defect, as this is a dominant strategy. Consider stage $N - 1$. Both players know that their opponent will defect at

stage N. Therefore, regardless of what actions are played at stage $N-1$, these actions will not impact the outcome of stage N and conversely the actions will also not lead to any punishment. Continuing this line of reasoning, it can be concluded that the optimal strategy for both players at each stage is (D, D) and this is also the Nash Equilibrium.

2.1.3.2 Infinitely Repeated Games

The infinitely repeated game of G has a payoff that is the discounted average

$$(1-\delta)\sum_{m=1}^{N}\delta^{m-1}u_i(a^m).$$

A key feature of infinitely repeated games is that there is no final stage. Therefore, there is always a threat of future punishment for both players. This is in stark contrast to finitely repeated games where it was demonstrated in the *Prisoner's Dilemma* example that reputation did not matter as neither player could punish the other in the future. There are several strategies that can be employed in the infinitely repeated *Prisoner's Dilemma*, and each strategy relies on the threat of future punishment to influence the other players move at the current stage [15].

The first strategy to be considered is Grim trigger. This strategy is defined as follows: $s_i(\emptyset) = C$ and

$$s_i(a^1, ..., a^n) = \begin{cases} C & \text{if } (a_j^1, \ldots, a_j^n) = (C, \ldots, C) \\ D & \text{otherwise} \end{cases} \tag{8}$$

for every history $(a^1, \ldots, a^n)$, where j is the other player. In words, player i will cooperate at each stage until player j defects. Once player j defects, player i will defect forever going forward. The main idea is that any short-term gain that one of players can receive for defecting at stage m will be negligible if the discount factor δ is sufficiently close to 1. This is because the other player will defect for all time after and the most the defecting player can achieve is a payoff of 1. Therefore, if both players select a Grim trigger strategy when δ is sufficiently close to 1, they each have no unilateral incentive to deviate and thus are in an equilibrium.

Tit-for-tat is another strategy that will be considered. The strategy is as follows. Let n be the first period that player II chooses D. Then player I chooses D in period n+1, and continues to choose D until player II reverts to C. It can be shown that if each player uses *tit-for-tat* when δ is sufficiently close to 1, then *tit-for-tat* is a Nash equilibrium in the infinitely repeated game.

2.1.4 Bayesian Games

In many conflict-resolution scenarios, each decision maker can have uncertainty about the preferences or intentions of the other decision maker. An example of such a scenario is poker, where each player has some uncertainty about their opponent's cards. The available information each player has is their cards and this information can be used to make probabilistic inferences about the opponent's cards. Bayesian games are a branch of game theory that considers scenarios where each decision maker has some uncertainty about the underlying state of the world [24]. In these games, each decision maker can observe signals that are correlated to the state and use their observations to update their beliefs about the underlying state. *In the previously mentioned poker example, the state of the world is the opponent's card and the signal is the players own cards.*

A Bayesian game consists of i) a set of players ii) a set of states iii) a set of signals for each player iv) a set of actions for each player v) a belief about the state, which are after-the-fact probabilities on the state given signals observed for each player vi) a Bernoulli payoff function over pairs (a, ω) , where a is an action profile and ω is a state [25]. *Note the following. vNM preferences are preferences regarding lotteries represented by the expected value of a payoff function over deterministic outcomes. A Bernoulli payoff function is a payoff function over deterministic outcomes whose expected value represents vNM preferences.* In the situation where one decision maker is privy to the underlying state, these games are referred to as asymmetric information games. For illustrative purposes, consider Battle of the sexes (BoS), a complete information game. Now suppose that the row player has uncertainty about the game payoffs. This modified BoS game is an example

of an asymmetric information game. It can be useful to interpret the game as the row player having uncertainty about the *type* of column player they face. Example payoff matrices for the BoS with two types of column player are

	B	S
B	$2,1$	$0,0$
S	$0,0$	$1,2$

State α

	B	S
B	$2,0$	$0,2$
S	$0,1$	$1,0$

State β (9)

where α and β are the types of the column player.

A **Bayes-Nash** equilibrium is the Nash equilibrium of the Bayesian game [16]. Note that a Bayesian game can be formulated as a game with complete information. In the complete information formulation, each type of column player is modeled as a distinct player. Therefore, if there are N types of column players in the Bayesian game, there will be $N + 1$ distinct players in the complete information game formulation.

2.2 Dynamic Programming

Dynamic programming describes the process of solving multistage decision problems, where the goal is to find the best decisions in succession that minimize a certain cost [26], [27], [28]. An important characteristic of these decision problems is that decisions at each stage cannot be viewed in isolation because of the need to trade off short-term and long-term costs. The dynamic programming method captures this tradeoff by breaking the decision problem into smaller subproblems [29]. At each stage, decisions are ranked based on the sum of the present cost and the expected future cost, assuming optimal decision making for subsequent stages. *Richard Bellman pioneered this idea, which is referred to as the* ***Principle of Optimality*** *[28].*

The systems that will be considered in this dissertation are discrete-time dynamic systems that have a cost function that is additive over time. *Notation in this chapter will be*

consistent with that of Bertsekas in [26]. This work ([26]) provides more in-depth coverage of the Dynamic Programing material. The general form of the system is

$$x_{k+1} = f_k(x_k, \mu_k, \omega_k), \quad k = 0, 1, \ldots, N-1$$

where k indexes discrete time, x_k is the state of the system, μ_k is the decision variable to be selected at time k, and ω_k is the disturbance. The total cost can be expressed as

$$g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k, \omega_k), \tag{10}$$

where $g_N(x_N)$ is a terminal cost incurred at the end of the process.

2.2.1 Imperfect State Information Problems

In a dynamic system, it often happens that there can exist uncertainty about the exact value of the current state x_k. This uncertainty can be attributed to the inaccessibility of some state variables, sensor error, or the cost of obtaining the exact value of the state [30]. These situations are modeled by assuming that the controller receives some observations about the value of the current state x_k at stage k. Problems where observations are used in place of the state are called **imperfect state information problems**. It turns out that imperfect state information problems can be reformulated as a perfect state information problems [26]. *Therefore, tools and approaches that are used to solve perfect state information problems can be also used to solve these problems.* An example problem will be presented next to demonstrate this reformulation.

Consider the following problem where the controller only has access to observations z_k of the form

$$z_0 = h_0(x_0, v_0), \quad z_k = h_k(x_k, \mu_{k-1}, v_k), \quad k = 1, 2, \ldots, N-1$$

where v_k is the observation disturbance and is characterized by a given probability distribution

$$P_{v_k}(\cdot \mid x_k, \ldots, x_0, \mu_{k-1}, \ldots, \mu_0, \omega_{k-1}, \ldots, \omega_0, v_{k-1}, \ldots, v_0).$$

Denote I_k to be the information available to the controller at time k, where

$$I_k = (z_0, z_1, \ldots, z_k, \mu_0, \mu_1, \ldots, \mu_{k-1}), \quad k = 1, 2, \ldots, N-1$$

The cost function of this problem is then expressed as

$$J_\pi = \mathbf{E}\left[g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(I_k), \omega_k) \right]_{x0, \omega_k, v_k,} \tag{11}$$

where each function μ_k maps the information vector I_k into the control space C_k and

$$\mu_k(I_k) \in U_k \;\; \forall I_k, k = 0, 1, \ldots, N-1$$

To reformulate the problem as a perfect information problem, it is necessary to define a new system whose state at time k is the set of all information I_k that the controller has available. Note that $I_{k+1} = (I_k, z_{k+1}, \mu_k)$. Therefore,

$$P(z_{K+1} \mid I_k, \mu_k) = P(z_{k+1} \mid I_k, \mu_k, z_0, z_1, \ldots, z_k).$$

Observe that the probability distribution of z_{k+1} depends explicitly on the state I_k and control μ_k of the new system. The cost function can be reformulated by writing

$$\mathbf{E}\left[g_k(x_k, \mu_k, \omega_k) \right] = \mathbf{E}\left[\mathbf{E}\left[g_k(x_k, \mu_k, \omega_k) \mid I_k, \mu_k \right]_{x_k, \omega_k} \right]$$

that leads to the new cost function

$$\tilde{g}_k(I_k, \mu_k) = \mathbf{E}\left[g_k(x_k, \mu_k, \omega_k) \right]_{x_k, \omega_k}$$

The problem has now been reformulated as a perfect information problem.

2.2.2 Approximate Dynamic Programming

Obtaining an optimal policy from the dynamic programing (DP) algorithm can often be computationally prohibitive. In many instances, as the size of the DP increases, the computational complexity increases exponentially; this phenomena is called the *"curse of dimensionality [28]." In fact, even for a perfect state information problem with Euclidean state*

and control spaces, DP can only be applied numerically if the dimensions of the spaces are relatively small [26]. Since imperfect state problems are more complex than their perfect state counterparts, only in special cases can a numerical solution be computed. To address the complexity issues that DP presents, methods for computing suboptimal policies are considered in the ensuing sections. Depending on the problem, some methods can be more appropriate than others in the since that those methods strike a better balance between convenient implementation and adequate performance. The methods that will be discussed in this section are limited lookahead, policy improvement, and model predictive control as these methods play a significant role in the derivation of the main theoretical results of this dissertation.

2.2.2.1 Limited Lookahead

Given an N-stage decision problem, the DP algorithm requires computation of optimal policies for stages $k = k' + 1, k' + 2, \ldots, N$ to evaluate the performance of the policies at stage $k = k'$. This computation is expensive and becomes increasingly prohibitive as N grows large. An approach to simplify the computation needed to evaluate the policies at stage k' is to truncate the time horizon and use at each stage a decision based on lookahead of a small number of stages [31]. The simplest approach is to consider looking ahead only one step into the future.

Let $\bar{\mu}_k(x_k)$ denote the control at stage k and state x_k that implements a one-step lookahead policy. The expression

$$\min_{u_k \in U_k(x_k)} \mathbf{E}\left[g_k(x_k, u_k, \omega_k + \tilde{J}_{k+1}(f_k(x_k, u_k, \omega_k))\right] \tag{12}$$

then represents the minimum cost for such a policy where $\tilde{J}_{k+1}$ is an approximation of the actual cost-to-go function J_{k+1}. Similarly, one can consider policies that look a fixed m steps into the future or consider a policy where the lookahead horizon recedes over time. The following theorem presented in [26] provides conditions under which the one-step lookahead policy achieves a cost $\bar{J}_k(x_k)$ that is better than the approximation $\tilde{J}_k(x_k)$.

Theorem 2 *[26] Assume that for all x_k and k, we have*

$$\min_{u_k \in \bar{U}_k(x_k)} \mathbf{E}\Big[g_k(x_k, u_k, \omega_k) + \tilde{J}_{k+1}(f_k(x_k, u_k, \omega_k))\Big] \leq \tilde{J}_k(x_k). \tag{13}$$

Then the cost-to-go functions $\bar{J}_k$ corresponding to a one-step lookahead policy that uses $\tilde{J}_k$ and $\bar{U}_k(x_k)$ satisfy for all x_k and k

$$\bar{J}_k(x_k) \leq \min_{u_k \in \bar{U}_k(x_k)} \mathbf{E}\Big[g_k(x_k, u_k, \omega_k) + \tilde{J}_{k+1}(f_k(x_k, u_k, \omega_k))\Big]. \tag{14}$$

It is worth pointing out that incrementing the lookahead steps can increase the performance of the policy, but it can also increase the computation substantially. Therefore, a key design decision is determining how many lookahead steps into the future is best. For a given $\tilde{J}_k(x_k)$ and a given process, there can be a significant diminishing return on performance as the lookahead horizon increases. *The discussion has been centered around applying this policy to finite horizon problems, but this policy is equally applicable/relevant to infinite horizon problems.*

2.2.2.2 Policy Improvement (Rollout policies)

Let a suboptimal policy $\pi = \{\mu_0, \ldots, \mu_{N-1}\}$ be referred to as a *base policy*. *Note that the base policy is typically a policy that is easily implementable.* Then the rollout algorithm can be viewed as a single step of the classical policy iteration method that starts from the base policy and yields an improved policy called the rollout policy [32]. *Alternatively, one can view the rollout policy as a one-step lookahead policy, with the optimal cost-to-go approximated by the cost-to-go of the base policy.* The policy improvement mechanism of the underlying policy iteration process generally allows rollout algorithms to magnify the effectiveness of any given heuristic through sequential application [33].

To speed up the computation of the rollout policy, one can restrict the set of controls to a subset $\hat{U} \subset U$. Also, one can consider an approximation $\hat{J}_{k+1}$ of $\tilde{J}_{k+1}$ to simplify computation. These modifications yield a minimization of the form

$$\min_{u_k \in \hat{U}_k(x_k)} \mathbf{E}\Big[g_k(x_k, u_k, \omega_k + \hat{J}_{k+1}(f_k(x_k, u_k, \omega_k))\Big] \tag{15}$$

that is more computationally tractable then the original minimization problem. The improvement in the speed of computation, however, generally comes at the expense of performance.

2.2.2.3 Model Predictive Control

Model predictive control (MPC), also known as receding horizon control, is a form of control in which the current action is determined by solving a finite horizon open-loop optimal control problem online at each sampling instant [34]. MPC combines elements of several ideas, rollout algorithms and limited lookahead, that have been discussed in the previous sections. The general process for MPC can be summarized as the following:

1. Obtain estimates of the current state of the system
2. Calculate optimal input minimizing the desired cost function over the prediction horizon using the system model and the current state estate for prediction
3. Implement the first part of the optimal input and discard the remaining parts of the input
4. Continue with 1.

Note that a key difference between MPC and conventional control is that conventional control uses a pre-computed control law [35]. A key point to emphasize is that although an open loop control problem is being solved at each time step k, there is still an implicit feedback mechanism. This mechanism is the new state x_{k+1}, which embodies all current and relevant information about system, that is fed back into the control problem to compute control actions for time step $k + 1$.

A key objective of MPC is to obtain a stable closed-loop system. However, it can be the case that the implementation of this control approach drives the closed-loop system outside the feasible region. Consider the following as an example. Suppose that MPC is used to compute control actions μ_k at time k and those actions are applied to the system. Since

the horizon of the control problem solved at time k is based on a prediction, a disturbance during time k can make the optimization problem infeasible. As a consequence, the system will then be unstable if control action μ_k is applied. Therefore it is important to analyze under what conditions will the application of MPC make the system unstable. This example highlights another issue that is the length l of the horizon to consider. Considering a longer horizon does not necessarily guarantee improved performance. In fact, it can happen that a shorter horizon improves the system's performance.

CHAPTER 3

REPEATED GAMES WITH ASYMMETRIC INFORMATION

3.1 Introduction

Repeated incomplete information games are a branch of game theory that considers the scenario where each decision maker has private information about the state of the world and must decide how to best use that information throughout a period of repeated interactions with other decision makers. Of interest in this research is the asymmetric case, where only one decision maker has private information. Since all other information is assumed public, private information implies superior information in this context. The dilemma faced by the informed decision maker is how to trade off the short-term reward versus long-term consequences for exploiting her[1] private information, since exploitation can reveal the true state of the world to the other decision makers; in the zero-sum games we consider, revelation costs her in the long-run. Seminal work by Aumann and Maschler [10] derives a recursive formula for the value of the zero-sum game, which quantifies the exploitation tradeoff, and also derives the optimal policy for the informed decision maker. Using Aumann and Maschlers' model for explicit computations of optimal policies is prohibitive for games with multiple stages.

3.1.1 Previous Work

Previous work to address complexity issues of computing optimal policies has been mostly limited to special cases of the simplest zero-sum games. Heur [36] derives optimal polices for a specific 2×2 matrix game with two states. Domansky and Kreps [5] expand on the work of Heur by considering a subclass of 2×2 matrix games with two states, where Heur's zero-sum game is contained within that subclass. They show that the games in their subclass satisfy a special condition, which allows them to derive optimal policies for the

[1] In this chapter, we will refer to the informed player as "she" and the uninformed player as "he". The assignment of "he" and "she" was arbitrary and was done for the purpose of clarity.

games. There is work that considers using suboptimal strategies to address the complexity issue for all classes of games [6]. In this work, the informed player never uses his information throughout the game, and accordingly is "non-revealing." While the suboptimal strategies are readily computable, only under special circumstances do these strategies offer strong suboptimal payoffs. In contrast to the previously mentioned work, Gilpin and Sandholm [37] consider computing optimal policies for the infinite horizon games by using a discritization technique to approximately solve a non-convex optimization problem.

3.1.2 Contributions of the Work

We address the complexity issues for both finite horizon and infinite horizon zero-sum repeated games by deriving a suboptimal policy based on the concept of policy improvement. *Policy improvement is a dynamic programming technique that first considers an initial policy, referred to as a baseline policy, that is easily implementable. Through an iterative process, the performance of the baseline policy is compared to that of other policies from a set of candidate policies. If a candidate policy performs better, it becomes the new baseline policy.* The baseline policy we consider is a non-revealing policy, i.e., one that completes ignores superior information. The improved policy, which is implemented in a receding horizon manner, strategizes for the current stage while assuming a non-revealing policy for future stages. We derive bounds on the guaranteed performance of the improved policy and establish that the bounds are tight. We show that the performance of the policy improvemement methods converge asymptotically to optimal with respect to the number of stages in the game. *As a result of the convergence result, policy improvement is optimal for infinitely repeated zero-sum games.* Last, we show that the improved policy can be computed by solving a linear program whose complexity is constant with respect to the game length.

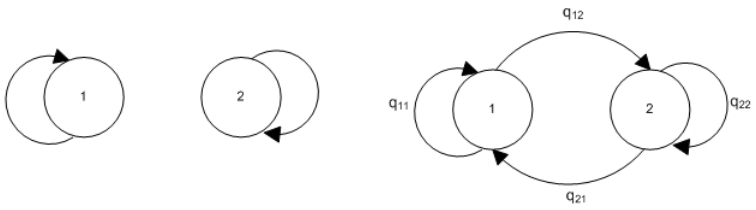
3.1.3 Outline

The outline for the rest of this chapter is as follows. In Section 3.2, we discuss fundamental zero-sum game definitions and concepts. This discussion will include both the repeated and

stochastic zero-sum games. Next, in Section 3.3, we present Aumann and Maschler's dynamic programming formulation for deriving optimal policies in zero-sum repeated games, and discuss the complexity issues of this formualtion. In Section 3.4, we discuss previous work on addressing the complexity issues of computing optimal policies. We then present the main results of this chapter in Sections 3.5 and 3.6. In Section 3.5, we introduce policy improvement methods to compute suboptimal strategies and derive bounds on the performance of the methods. In Section 3.6, we show how the policy improvement methods can be computed by solving a LP whose complexity is invariant with respect to the game length. Simulations that demonstrate the performance of the policy improvement methods are presented in Section 5.8. Last, we conclude with a brief summary of the chapter.

3.2 Preliminaries on Zero-sum Games with Asymmetric Information

Optimal information exploitation is a key issue in zero-sum games with asymmetric information. In these games, player I is informed about the current state of the world at each stage, while player II is not. The simplest case, repeated games, is where the state of world is chosen by nature before the initial stage of the game and remains fixed over all stages. *An illustration of a two state repeated game is depicted in Figure 1a.* The general case, stochastic games, is where the state of the world can change at each stage. *An illustration of a two state stochastic game where the state transitions are independent of the player's actions is depicted in Figure 1b.* In either case, each player's knowledge about the other's past actions changes over time, affects their beliefs, and introduces a dynamic aspect to the games. Specifically, player II can use his observations of player I's past actions to build beliefs about the current state of the world. Therefore, player I must strategically select an appropriate action at each stage that transmits the desired information or misinformation to player II. In this section, we will present models, definitions, and concepts for both repeated and stochastic games. We will also present examples that illustrate the information exploitation issues that player I experiences.



(a) Repeated Game **(b) Stochastic Game**

Figure 1: Repeated games are a special case of stochastic games where all states are said to be absorbing.

3.2.1 Repeated Games

Two players repeatedly play a zero-sum matrix game over stages $m = 1, 2, ..., N$. The row player is the maximizer, and the column player is the minimizer. The specific game is selected from a finite set of possible games (or states of the world), K. Let $\Delta(B)$ denote the set of probability distributions over some finite set, B. Define S to be the set of pure actions of the row player and similarly define T to be the set of pure actions of the column player. The game matrix at state $k \in K$ is denoted $M^k \in \mathbb{R}^{|S|\times|T|}$. Before the initial stage $m = 1$, nature selects the specific game according to a probability distribution $p \in \Delta(K)$, which is common knowledge. The outcome of this selection is not revealed to the uninformed player. Once selected, the game remains fixed over all stages. The n-stage game is denoted as $\Gamma_n(p)$. Similarly, the discounted and infinitely repeated games are denoted as $\Gamma_\lambda(p)$ and $\Gamma_\infty(p)$ respectively.

3.2.1.1 Strategies

For $n = 1, 2, \ldots,$ let H_n=$[S \times T]^{n-1}$ be the set of possible histories at stage n. Then $h_n \in H_n$ is a sequence $(s_1, t_1; s_2, t_2; \ldots; s_{n-1}, t_{n-1})$ of moves of the two players in the first $n-1$ stages of the game. Let $X = \Delta(S)$ and $Y = \Delta(T)$ denote the mixed moves of player I and player II respectively where $x_n^k = (x_n^k(s))_{s\in S}$. Let $\zeta_n : k \times h_n \mapsto S$ denote a pure strategy for player I and define $\zeta = (\zeta_1, \zeta_2, \ldots, \zeta_n)$. Similarly let $\psi_n : h_n \mapsto T$ denote a pure strategy of player II and define $\psi = (\psi_1, \psi_2, \ldots, \psi_n)$. Mixed strategies are probability distributions

over pure strategies. Behavior strategies are sequences of mappings from $K \times H_n$ to X and mappings from H_n to Y for player I and player II respectively. Aumann [8] showed that mixed strategies can be equivalently represented as behavior strategies for the zero-sum games considered in this paper. Therefore, the terms behavior strategy and mixed strategy will be used interchangeably. Let σ and τ denote the behavior strategies of players I and II respectively. Also let $\sigma(h_n) = (x_n^k)_{k \in K} \in X$ be defined as the vector of mixed moves of player I at stage n.

3.2.1.2 Payoffs

Let $M_{i,j}^k$ denote element (i, j) of payoff matrix M^k and also denote $g_m = M_{i_m, j_m}^k$ to be the random payoff at stage m. Let $\gamma_m^p(\sigma, \tau)$=$\mathbf{E}\left[g_m\right]_{p,\sigma,\tau}$ denote the expected payoff for the pair of behavioral strategies (σ, τ) at stage m. The payoff for the n-stage game is then defined as

$$\bar{\gamma}_n^p(\sigma, \tau) = \frac{1}{n} \sum_{m=1}^{n} \gamma_m^p(\sigma, \tau). \tag{16}$$

Similarly the payoff for the λ-discounted game is defined as

$$\bar{\gamma}_\lambda^p(\sigma, \tau) = \sum_{m=1}^{\infty} \lambda(1-\lambda)^{m-1} \gamma_m^p(\sigma, \tau). \tag{17}$$

Definition 1 *Player I can guarantee $\phi \in R$ in the game $\Gamma_n(p)$ if, for every $\epsilon > 0$ there exists a strategy σ of player I and $N \in \mathbf{N}$ such that*

$$\forall \tau, \forall n \geq N, \ \bar{\gamma}_n^p(\sigma, \tau) \geq \phi - \epsilon$$

Definition 2 *Player II can defend $\phi \in \mathbf{R}$ in the game $\Gamma_n(p)$ if, for every $\epsilon > 0$ and every strategy σ of player* 1*, there exists a strategy τ of player II and $N \in \mathbf{N}$ such that*

$$\forall n \geq N, \ \bar{\gamma}_n^p(\sigma, \tau) \leq \phi + \epsilon.$$

3.2.1.3 Beliefs

Since player II is not informed of the selected state k, he can build beliefs on which state was selected. These beliefs are a function of the initial distribution p and his observed

moves of player I. Therefore, player I must carefully consider her actions at each stage as they could potentially reveal the true state of the world to player II. In order to get a worse case estimate of how much information player I transmits through her moves, she models player II as a Bayesian player and assumes that player II knows her behavior strategy. The updated belief p^+ is computed as

$$p^+(p, x, s) = \frac{p^k x^k(s)}{\bar{x}(p, x, s)}, \tag{18}$$

where $\bar{x}(p, x, s) := \sum_{k \in K} p^k x^k(s)$.

3.2.1.4 Non-revealing Strategies

Revealing information is defined as using a different mixed strategy in each state k at stage m. From (69), it follows that a mixed strategy x_m at stage m does not change the current beliefs of the row player if $x_m^k = x_m^{k'} \; \forall \; k, k'$. As a consequence, not revealing information is equivalent to not changing the column player's beliefs about the true state of the world. In stochastic games, it is possible for the column player's beliefs to change even if the row player uses an identical mixed strategy for each state k.

An optimal non-revealing strategy can be computed by solving

$$u(p) = \max_{x \in \mathrm{NR}} \min_{y} \sum p^k x^k M^k y, \tag{19}$$

where the set of non-revealing strategies is defined as $\mathrm{NR} = \{x_m \,|\, x_m^k = x_m^{k'} \;\; \forall k, k' \in K\}$. By playing an optimal non-revealing strategy at each stage of the game, the row player can guarantee a game payoff of $u(p)$. Note that the optimal game payoff for the n-stage game, $v_n(p)$, is equal to $u(p)$ only under special conditions.

Definition 3 *Let Cav*$[u(p)]$ *be the point-wise smallest concave function g on* $\Delta(K)$ *satisfying* $g(p) \geq u(p) \; \forall p \in \Delta(K)$.

In words, $\mathrm{Cav}[u(p)]$ can be guaranteed for any repeated game with asymmetric information.

3.2.1.5 Signals

As discussed in Section 3.2.1.3, player II can build beliefs about the state of the world based on his observations of player I. Since each player's actions are observable by the other player, player I can use her actions in such a manner to influence player II's beliefs about the true state of the world. Player I can also influence her opponent's beliefs by introducing auxiliary signals into the game to communicate additional information or misinformation about the true state. These signal can be correlated with the state of the world and are announced by player I and observed by player II.

Let L be the set of auxiliary signals and let l be a signal contained within the set. Denote $\mu^k(l)$ to be the probability that player I selects signal l in state k. The following lemma provides a method that allows player I to construct a specific μ to change the beliefs of player II in a desired direction. *Note that the auxiliary signal is also a signal to player I and influences her move selection.*

Lemma 3.2.1 *[7] Let L be a finite set and* $p = \sum_{l \in L} \alpha_l p_l$ *with* $\alpha \in \Delta(L)$ *, and* $p, p_l \in \Delta(K)$ *for all l in L. Then there exists a* μ *that changes the beliefs of player II from* (K, p) *to L between the transition from the current stage and the next stage such that*

$$P(l) = \alpha_l \quad \textit{and} \quad P(\cdot|l) = p_l$$

where $P = p \cdot x$ *is the probability induced by p and* μ *on* $K \times L : P(k, l) = p^k \mu^k(l)$.

Proof: Let

$$\mu^k(l) = \alpha_l \frac{p_l^k}{p^k},$$

for k in the support of p. Then

$$P(l) = \sum_{k \in K} p^k \mu^k(l) = \alpha_l$$

and

$$P(k|l) = \frac{p^k \mu^k(l)}{\alpha_l} = p_l^k.$$

3.2.1.6 Examples

In this section, we will consider three examples that illustrate the ways player I can exploit her information. In the first example, there are two states of the world and each state is equally likely to be selected. Player I is told before the start of the game whether the state is α or β. The payoff matrices for this game are

	L	R
T	1	0
B	0	0

State α

	L	R
T	0	0
B	0	1

State β (20)

She has a choice of selecting move T or move B in each state. However, she favors selecting T in state α and B in state β because this is a dominant strategy. A standard approach to evaluate the performance of a given strategy is to assume player II is told the strategy. Suppose player I plays her dominant strategy. Upon seeing T, player II will immediately know the state is α. Similarly, upon seeing B, he will know the state is β. This is why the dominant strategy is said to be completely revealing in this example. After stage 1, player II will play R if state α and L in state β, which will net player I an expected stage payoff of 0 from stage 2 onward. The payoff for player I with respect to n would be $\frac{1}{2n}$ and so the immediate gain for exploiting information diminishes for games with large n.

Another possible strategy that player I can consider is a non-revealing strategy. In this strategy, player I plays as if she is oblivious of the state of the world. This situation is equivalent to playing the average matrix game $\bar{M}$ =

	L	R
T	$\frac{1}{2}$	0
B	0	$\frac{1}{2}$

(21)

where $\bar{M} = p^{\alpha}M^{\alpha} + p^{\beta}M^{\beta}$. A solution to the average game can be computed by solving $u(p)$. If player I plays T and B with equal probability independent of the state, she can

guarantee a payoff of $\frac{1}{4}$. Therefore player I is better off not using her information in the long-run.

The second example has the same parameters as the previous example, except for the payoff matrices that are

	L	R
T	-1	0
B	0	0

State α

	L	R
T	0	0
B	0	-1

State β (22)

If player I uses her dominant strategy, she will guarantee an expected payoff of 0. This dominant strategy, which is to play B in state α and T in state β, is completely revealing and optimal for all n. This is because the best payoff player I can achieve is 0 in either state. The average game payoff matrix is

	L	R
T	$-\frac{1}{2}$	0
B	0	$-\frac{1}{2}$

(23)

and the expected payoff for the average game is $-\frac{1}{4}$. Whereas not using information was the best long-term decision for player I in the first example, in this example exploiting information is optimal for the short and long term.

The last example we will consider has payoff matrices

	L	M	R
T	4	0	2
B	4	0	-2

State α

	L	M	R
T	0	4	-2
B	0	4	2

State β

Similarly to the previous examples, the other parameters of the game are the same as in the previous examples. Also similar to previous examples is that playing a dominant strategy

is completely revealing and achieves a payoff of 0 in the long-run. If player I decides to not use her information, she will be playing the non-revealing game

	L	M	R
T	0	4	-2
B	0	4	2

that also achieves a long-term payoff of 0. In all of the examples that we have presented so far, we have only considered completely revealing and non-revealing strategies. There is another strategy that player I can consider, which is a partially revealing strategy. The main idea of partially revealing strategies is that this strategy is partially correlated with the true state of the world. Instead of player I playing T in state α with probability 1 and B in state β with probability 1, she does the following. If the state is α play T with probability $\frac{3}{4}$ at stage 1. If the state is β, play B with probability $\frac{3}{4}$ at stage 1. Player I's mixed actions at stage $m = 1$ are then $\tilde{x}_1^1 = [.75, .25]$ and $\tilde{x}_1^2 = [.25, .75]$. From stage 2 onward, she will then play nonrevealing.

To evaluate the performance of this strategy, we will assume player II knows it. In the first stage player I can guarantee an expected stage payoff of 1 by playing $\tilde{x}$. Suppose that T is selected by player I at the first stage, then the beliefs of player II will be

$$p_T = [.75, .25] = p^+(p, \tilde{x}, T),$$

where his beliefs about state α are $\frac{3}{4}$. Similary players II's beliefs upon seeing B are

$$p_B = [.25, .75] = p^+(p, \tilde{x}, B).$$

By playing non-revealing from stage 2 going forward, players I's long-term payoff will either be $u(p_T)$ or $u(p_B)$. In this example

$$u(p_T) = u(p_T) = 1.$$

Therefore player I can guarantee an expected game payoff of at least 1, independent of what player II does. The idea of using signals, which in this example are players I's actions, to

change the beliefs of player II is the heart of asymmetric information games. This is also an idea we will exploit in the policy improvement methods we present later in the paper.

3.2.2 Remarks

The primary focus of this chapter is to address the challenges that an informed player faces with regard to information exploitation and computation of optimal policies. An interesting aspect of incomplete information games that we do not address is the challenges that an uninformed player faces learning the actual state of the world. We will devote this section to discussing these issues for repeated games. *Player II experiences similar issues in the stochastic games also.* We will also present a heuristic for computing a suboptimal strategy that is inspired by a central idea used to construct optimal policies for player II in the finite and infinite horizon games. This idea involves considering the actual payouts that player I would receive for each state of the world. Since player II can observe the moves of player I, he can construct a vector of payoffs, where the kth element corresponds to player I's payoff if the state was k. Player II's strategy is then to ensure that the payoff vector remains within some bounds. Heur in [36] discusses how to enforce bounds on vector payoffs for the game with payoff matrices

	L	R
T	3	-1
B	-3	1

State α

	L	R
T	2	-2
B	-2	2

State β

These results, however do not extend to the general n-stage game. In [6], a method to enforce vector payoffs is discussed that guarantees an optimal payoff to player I for the infinitely repeated games. This method incorporates Blackwell's approachability theorem to enforce payoffs and guarantee an expected payout of Cav[$u(p)$])

3.2.2.1 Challenges of learning the true state

Learning the true state of the world is challenging for player II because player I can play in a deceptive manner [36]. If player II new the mixed actions x^k that player I would use for each state k, then learning player I's type would be straightforward because the defender could follow the standard Bayesian update approach. Unfortunately, in the actual play of the game, he does not know her mixed strategy as this information is private. Another approach that the he can consider is solving a linear program to compute an optimal defensive strategy. However, the complexity of the LP is exponential with respect to n, the number of stages of the game. We will now introduce a payoff-based heuristic for learning player I's type that is computationally tractable for arbitrary n and only depends on the information that the defender has. This information is namely the history of player I's actions.

3.2.2.2 Heuristic

The main idea behind the payoff-based heuristic is as follows. Player II's belief of a player I of type k will be correlated with the actual game payoff of player I. After each stage, player II keeps track of what the overall game payoff would be for each type of player I. The game payoff for a player I of type k at stage n, given history h_n will be denoted $\tilde{\gamma}_n^k(h_n)$. The game payoff $\tilde{\gamma}_n^k(h_n)$ for a player I of type k will be compared to the best possible payoff that a type k player I can achieve. We will denote the best possible payoff by $|M^k|$, where $|M^k| := \max_{i,j} M_{i,j}^k$.[2] Let

$$\xi^k(h_n) \propto \frac{\tilde{\gamma}_n^k(h_n)}{|M^k|} \tag{24}$$

be a measure of the likelihood that an attacker is of type k given history h_n. The belief update procedure is then

$$\tilde{p}_{n+1}^k(h_n) = \tilde{p}_n^k \frac{\xi^k(h_n)}{\bar{\xi}(h_n)} \tag{25}$$

where $\bar{\xi}(h_n) = \sum_{k \in K} \tilde{p}_n^k \xi^k(h_n)$. To compute a best response strategy $\tilde{y}^*$ for player II given the

[2]Without loss of generality, we will assume in this section that each matrix M^k is scaled with values ranging from 0 to 1.

approximate belief $\tilde{p}_n$ at stage m, solve the optimization equation

$$\tilde{y}_m^* = \arg\min_{y_m} \max_{x_m} \sum_{k \in K} \tilde{p}_m^k x_m^k M^k y_m. \tag{26}$$

The goal of the heuristic is to allow player II to estimate the probability that player I is of a particular type by using only the information available during the play of the game. This information is namely the history of moves and the payoff matrices. *Recall that in order for player II to perform Bayesian inference, player II needs to know the mixed strategy of player I. In the actual play of the game, player II does not have access to this information.* By doing an analysis of the payoffs of each type of player I given the current history, player II can observe which types currently have the best average payoffs up to the current stage. *The underlying assumption behind the heuristic is that the types that are achieving the best possible payoff given their type are more likely to be the opponent player II is facing in the game.* Once player II computes an estimate of player I's type using the heuristic, player II can then play a best response given current beliefs.

3.3 Dynamic Programing Formulation for Optimal Policies

3.3.1 Finite Horizon Games

The seminal work of Aumann and Maschler [10] introduced a dynamic programing formulation

$$v_{n+1}(p) = \frac{1}{n+1}\left[\max_{x_1} \min_{y_1} \sum_{k \in K} p^k x_1^k M^k y_1 + n \sum_{s \in S} \bar{x}_s v_n\Big(p^+(p, x_1, s)\Big)\right], \tag{27}$$

that characterizes the value of n-stage zero-sum repeated games with asymmetric information. This formulation can also be interpreted as modeling player I's tradeoff between the short-term gain and the long-term informational advantage. Consider the case where $n = 1$ as a first example to illustrate the exploitation tradeoff. The formulation for $n = 1$ is then

$$v_1(p) = \max_{x_1} \min_{y_1} \sum_{k \in K} p^k x_1^k M^k y_1. \tag{28}$$

Since this is a one-shot game, there are no future consequences for exploiting information at the first stage. Therefore playing greedy is optimal. In games where $n \geq 2$, $v_n(p)$ is a

Bellman's equation with a stage cost of $\sum_{k\in K} p^k x_1^k M^k y_1$ and a cost-to-go of

$$n \sum_{s\in S} \bar{x}_s v_n\Big(p^+(p, x_1, s)\Big).$$

The cost-to-go function characterizes how information that is exploited by player I at the current stage will cost him in the future. It is the cost-to-go function that makes explicit computation of $v_n(p)$ prohibitive. This is because this function is recursive, and as a consequence, the number decision variables grows exponentially with respect to game length. One of the goals of this research is to present policy improvement methods to compute suboptimal polices that have a computational complexity that is invariant with respect to the length of the game.

Aumann and Maschlers formulation has the following bound

$$v_n(p) \geq \text{Cav}[u(p)].$$

As a a result, player I can guarante a payoff of Cav$[u(p)]$ for the n-stage game. Another result of Aumann and Maschler is the rate that the n-stage game converges to Cav$[u(p)]$. Specifically, the convergence of $v_n(p)$ to Cav$[u(p)]$ is

$$\text{Cav}[u(p)] \leq v_n(p) \leq \text{Cav}[u(p)] + \frac{C}{\sqrt{n}} \sum_{k\in K} \sqrt{p^k(1-p^k)}$$

where $C = \max_{i,j,k} M_{i,j}^k$.

3.3.2 Infinite Horizon Games

Similar to n-stage games, the dynamic programming formulation for the λ-discounted repeated game is

$$v_\lambda(p) = \max_{x_1} \min_{y_1} \lambda \sum_{k\in K} p^k x_1^k M^k y_1 + (1-\lambda) \sum_{s\in S} \bar{x}(s) v_\lambda\Big(p^+(p, x_1, s)\Big). \tag{29}$$

A relationship between the infinitely repeated games $v_\infty(p)$ and $v_\lambda(p)$ is the following.

Lemma 3.3.1 *[6] $v_\infty(p)$ and $v_\lambda(p)$ converge uniformly (as $n \to \infty$ and $\lambda \to 0$, respectively) to the same limit, Cav$[u(p)]$, which can be defended by player II in the infinitely repeated game.*

Also similar to the n stage game, the bound on the rate that v_λ converges to Cav$[u(p)]$ is

$$\text{Cav}[u(p)] \leq v_\lambda(p) \leq \text{Cav}[u(p)] + C\sqrt{\frac{\lambda}{2-\lambda}}\sum_{k\in K}\sqrt{p^k(1-p^k)}$$

3.4 Previous Work

3.4.1 Finite Horizon Repeated Games

Heur [36] analyzed the game with payoff matrices

	L	R
T	3	-1
B	-3	1

State α

	L	R
T	2	-2
B	-2	2

State β (30)

and showed how to explicitly compute optimal strategies for both players in this game. Domansky and Kreps [5] considered two-state zero-sum games with matrices M^α and M^β that represents payoffs for states α and β respectively. Domansky and Kreps provide optimal strategies for games where the condition

$$\text{val}[pM^1 + (1-p)M^2] = p\text{val}[M^1] + (1-p)val[M^2] \tag{31}$$

holds. *Note that val[M] denotes the value of a zero sum matrix game M.* They show that (31) holds for games with mixed-type payoff structure

	L	R
T	a	0
B	0	$-b$

State α

	L	R
T	$-\lambda a$	0
B	0	λb

State β (32)

and saddle-point type payoff structure

	L	R
T	1	0
B	0	$-(1-a)$

State α

	L	R
T	-1	0
B	0	$(1-a)$

State β (33)

It happens that (30) is an example of a saddle-point type game. Although Domansky and Kreps expands upon the work of Heur by generalizing optimal-policy computation results for a subclass of 2×2 matrix games, this subclass comprises only a small portion of the general class of 2×2 matrix games.

Zamir [38] considers classifying games in terms of the rate at which the games converges from v_1 to v_∞ and then constructing essentially-optimal policies whose payoff also converge at such a rate. As an example, Zamir showed that the game with payoff matrices

	L	R
T	1	0
B	0	-0

State α

	L	R
T	0	0
B	0	1

State β

has a convergence rate of $O(\frac{log(n)}{n})$. Knowing the convergence rate of this game suggests a method to derive an essentially optimal policy. Zamir notes, however, that determining the convergence rate for an arbitrary game is an open question. Therefore, deriving essentially optimal methods is challenging in general.

3.4.2 Infinite Horizon Repeated Games

Gilpin and Sandholm [37] consider infinitely-repeated zero-sum games with asymmetric information. They first present an optimization problem

$$\max_{p,\alpha} \sum_{l \in L} \alpha_l u(p_l)$$

$$\text{s.t.} \quad \sum_{l \in L} \alpha_l p_l = p \tag{34}$$

where

$$p^l \in \Delta(K) \text{ for } l \in L \text{ and } \alpha \in \Delta(L)$$

for computing Cav$[u(p)]$. Gilpin et. al then apply basic convex analsysis results to show that it is sufficient to set $|L| = K + 1$ to compute Cav$[u(p)]$. Next they approximately solve (34) by using discretization methods. Last, they show how optimal strategies for player I for the infinitely repeated games can be derived from the solved optimization problem (34). Specifically, let α^* and $p_l^* \in \Delta(K)$ be solutions to (34), then player I's optimal strategy is to choose l with probability $\frac{\alpha_l^* p_l^{k^*}}{p^k}$ and play the optimal mixed actions for $u(p_l)$.

3.4.3 Receding Horizon Control

In this section we will briefly summarize the main idea of receding horizon optimization and then proceed to discuss applications of this optimization technique in game settings. We will begin by considering the following Bellman's equation

$$J(x_0) = \max_{a_0 \in \Gamma(x_0)} \{g(x_0, a_0) + \beta J(T(x_0, a_0))\}$$

where x_0 denotes the initial state of the system, J denotes the system's performance, a_0 is the control input for state x_0, and T is the transition function that maps the current input and state to the next state. Observe that if the cost-to-go function is difficult to compute, then computing the optimal cost can be prohibitive. In situations where the optimal cost J is difficult to compute, an approximate Bellman's equation of the form

$$\bar{J}(x_0) = \max_{a_0 \in \Gamma(x_0)} \{g(x_0, a_0) + \beta \tilde{J}(T(x_0, a_0))\} \tag{35}$$

can be considered, where $\tilde{J}$ is an approximate cost-to-go function that can be readily computed.

Given the above setup, receding horizon optimization can be leveraged to derive sub-optimal controls for the system at each state. The process of receding horizon optimization

can be summarized as follows. First, obtain the estimate of the current state x of the system. Next, calculate the optimal input maximizing the desired objective function. Then, implement the first part of the input a and discard the remain parts of the input. Next, continue with the first step.

Receding Horizon optimization has been applied to various game settings. Cruz et. al. [39] studied a military operation game with one and two step receding horizon. Van den Broek [40] considered applying a receding horizon approach for non zero-sum differential games. Chang and Markus [41] consider a receding horizon approach for two-person zero-sum Markov Games. In their work, the minimizing player selects a "small" horizon and solves the game with the finite horizon (called the subgame) under the assumption that the maximizer makes her decision based on her best performance of the subgame.

3.5 Policy Improvement Methods

Policy improvement is a dynamic programing approach that consists of first considering an initial policy whose performance is known or can be readily computed. *We will select a non-revealing policy, i.e. one that ignores superior information, as our initial policy in this section and will refer to this policy as the baseline policy.* Next, a candidate policy is selected from some policy set. If the current candidate policy performs better than the baseline policy, the candidate policy becomes the new baseline policy. This new baseline policy is an improvement over the previous baseline policy, and therefore a policy improvement. This process can then repeated in an iterative manner.

The policy improvement methods presented in this section are based on the ideas of receding horizon optimization. In the n-stage game, player I solves an optimization problem over the interval $[m, n]$, where m is the current stage and n is the total number of stages. We assume that in the future stages, she will only consider non-revealing strategies. This assumption is key to reducing the complexity of the original dynamic programing formulation $v_n(p)$ because of the following idea. By playing non-revealing in all future stages,

player II's beliefs $\tilde{p}_{m+1}$, $\tilde{p}_{m+2}$, ..., $\tilde{p}_n$ at stages $m+1, m+2, \ldots, n$ respectively, remain constant. As a consequence, her expected future payout over the interval $[m+1, n]$ can be computed exactly by solving the optimization problem $u(\tilde{p})$. Similarly in infinite horizon games, the expected stage payout for all future stages is $u(\tilde{p})$, assuming player I plays non-revealing over all stages in the future. Note that $u(\tilde{p})$ can itself be computed by solving a linear program whose complexity is invariant with respect to n.

3.5.1 Finite Horizon

3.5.1.1 One-time policy improvement

In one-time policy improvement, player I strategizes for the *first stage* of the game while assuming that she will play in a non-revealing manner in all future stages. A dynamic programming formulation for this policy is

$$\tilde{v}_n(p) = \max_{x,\mu} \min_{y} \Big[\frac{1}{n} \sum_{k \in K} \sum_{l \in L} \mu^k(l) p^k x^k M^k y + \frac{n-1}{n} \sum_{s \in S} \sum_{l \in L} \bar{x\mu}(s,l) u(\tilde{p}^+(p, \mu, x, s, l)) \Big]$$

where $\bar{x\mu}(s,l) = \sum_{k \in K} p^k x^k(s) u^k(l)$ and the kth component of $\tilde{p}^+$ is

$$\frac{p^k \mu^k(l) x^k(s)}{\sum_{k \in K} p^k x^k(s) \mu^k(l)}$$

for the n-stage game. We will show that $\tilde{v}_n(p)$ guarantees a payoff to player I of Cav$[u(p)]$.

Theorem 3 *One-time policy improvement guarantees a payoff of at least Cav*$[u(p)]$ *for the n-stage repeated game.*

Proof: First note that Cav$[u(p)]$ can be expressed as $\sum_{l \in L} \alpha_l u(p_l)$, where $\alpha \in \Delta(L)$, $|L| < \infty$, $p, p_l \in \Delta(K)$, and $p = \sum_{l \in L} \alpha_l p_l$. Also note that Cav$[u(p)]$ can be computed by solving the following optimization problem

$$\max_{p,\alpha} \sum_{l \in L} \alpha_l u(p_l)$$

$$\text{s.t.} \quad \sum_{l \in L} \alpha_l p_l = p \tag{36}$$

where the constraint $\sum_{l\in L}\alpha_l p_l = p$ is nonlinear and nonconvex. *In this proof, we will show an equivalent representation of this problem that allows us to relax the nonlinear constraint. This relaxation is critical for a key result of this chapter, which is a linear programing formulation to compute policy-improvement strategies.* Observe that Equation 36 can be rewritten as

$$\max_{p,\alpha} \quad \left[\frac{1}{n}\sum_{l\in L}\alpha_l u(p_l) + \frac{n-1}{n}\sum_{l\in L}\alpha_l u(p_l)\right]$$

$$\text{s.t.} \quad \sum_{l\in L}\alpha_l p_l = p \tag{37}$$

where $n \geq 1$. Next, through algebraic manipulation (37) can be equivalently expressed as

$$\begin{aligned}\max_{x,\mu}\min_{y}\Big[\frac{1}{n}\sum_{k\in K}\sum_{l\in L}\mu^k(l)p^k x^k M^k y + \frac{n-1}{n}\sum_{l\in L}\bar{\mu}(l)u\big(\tilde{p}^+(p,\mu,l)\big)\Big]\\ \text{s.t.} \quad \sum_{l\in L} p^k\mu^k(l) = p^k \;\forall\, k\\ x^k = x^{k'}\;\forall k,k'\end{aligned} \tag{38}$$

where $\bar{\mu}(l) = \sum_{k\in K}\mu^k(l)$. Observe that the constraint

$$\sum_{l\in L} p^k\mu^k(l) = p^k \;\forall\, k$$

in (38) can be relaxed because it is trivially satisfied and can then be equivalently expressed as

$$\begin{aligned}\max_{x,\mu}\min_{y}\Big[\frac{1}{n}\sum_{k\in K}\sum_{l\in L}\mu^k(l)p^k x^k M^k y + \frac{n-1}{n}\sum_{s\in S}\sum_{l\in L}x\bar{\mu}(s,l)(\tilde{p}^+(p,\mu,x,s,l))\Big]\\ \text{s.t.} \quad x^k = x^{k'}\;\forall k,k'\end{aligned} \tag{39}$$

by algebraic manipulation. *Observe that the constraints are linear in this formulation of the optimization problem.* Last note that the optimal value of $\tilde{v}_n(p)$ is at least that of (39) because the feasible set of (39) is a subset of $\tilde{v}_n(p)$.

A game theoretic interpretation of the previous result for the case where constraint $x^k = x^{k'} \forall k, k'$ is enforced, as in (38), is the following. In a one-time policy-improvement strategy, the mixed strategy that player I plays, is a function of the state k, which she is informed of prior to the first stage of the game. To learn the state k, player II can use her observation of actions and auxiliary signals to build beliefs since the actions and signals are correlated with the true state. When the constraint $x^k = x^{k'} \forall k, k'$ is enforced, player I's mixed actions do not transmit any information about the current state,but the signals can transmit information. Therefore it is sufficient for player II to consider only the signals as a means to infer the actual state of the world. With one-time policy improvement, at the first stage, player I decides on the signal probabilities $\mu^k(l)$ for each state k. She then randomly selects a signal based on $\mu^k(l)$ and then plays an optimal strategy of $u(p_l)$, denoted x_l^*, for all stages. From the perspective of player II, by Lemma 3.2.1, the best case is that he can observe signal l. His beliefs upon observation of l become p_l and the game that is being played at stage 1 is then $u(p_l)$. Since this is the only signal emitted in this game, then the game payoff for seeing l is then $u(p_l)$. Since the overall likelihood of player I selecting l based on the mixture $u^k(l)$ is α_l,where $\alpha_l = \bar{u}(l)$, then player I can guarantee an overall game payoff of $\sum_{l \in L} \alpha_l u(p_l)$. It then follows that if player I uses μ^*, she can guarantee Cav$[u(p)]$. She will play with the mixed actions x_{opt}^k at each stage where $x_{opt}^k = \sum_{l \in L} \mu^k(l) x_l^*$. Relaxing the constraint allows for a richer set of strategies for player I to be considered (i.e., a strategy where beliefs are dependent on signals and moves).

3.5.1.2 Perpetual policy improvement

Similar to one-time policy improvement, in the perpetual policy improvement method, player I strategizes for the first stage while assuming a non-revealing strategy in all future stages. In contrast, once player I arrives at the next stage $m' = m + 1$, she can reevaluate her decision to continue playing the game $u(p_{ls})$ based on the signal l and the action s she observed in the previous stage. She does this by setting $p = p_{ls}$ and implementing the one-time policy improvement strategy at stage m'.

Corollary 1 *Perpetual policy improvement guarantees a payoff of at least Cav*$[u(p)]$ *for the n-stage repeated game.*

Proof: We will prove by construction. First, consider stage $m = 1$ and implement a one-time policy-improvement strategy. By Theorem 3, there exists a mixed action x^* and a mixed signal μ^* that guarantees a stage payoff of at least Cav$[u(p)]$ and a future payoff of at least Cav$[u(p)]$. The future payoff can be expressed as $\sum_{l\in L} \alpha^*_{ls} u(p^*_{ls})$. Next consider stage $m = 2$. Suppose signal l and move s were observed in stage 1, then the belief are p_{ls} at stage 2 and the payoff for playing a non-revealing strategy is $u(p_{ls})$. If player I plays non-revealing over the next $n - 1$ stages, she can guarantee an expected payoff of $u(p_{ls})$. Suppose however that player I implements a one-time policy-improvement strategy at stage $m = 2$. Note a key observation, which is that

$$\tilde{v}_n(p) \geq u(p) \; \forall n \geq 1, k \in K.$$

This implies that

$$\tilde{v}_{n-1}(p^*_{ls}) \geq \sum_{m=2}^{n} \frac{1}{n-1} u(p^*_{ls}).$$

Therefore, the worst that can happen to player I if she decides to implement a one-time policy improvement at stage $m = 2$ is that she gets the same payoff $u(p^*_{ls})$ she would have received if she hadn't deviated from the original policy decided at stage $m = 1$. Consider stage m. By a similar argument, perpetual policy improvement up to stage m guarantees an expected payoff of at least Cav$[u(p)]$.

Algorithm 1 Perpetual policy improvement

procedure PERPPOLICYIMPROVE
  initialize: set $p_1 = p$
  for $m = 1 \rightarrow N$ **do**
    compute $\hat{x}_m$ by solving one-time policy improvement LP with p_m
    select a move s for attacker type k using mixed strategy $\hat{x}_m^k$
    update beliefs vector (i.e.$p_{m+1} = p^+(p, \hat{x}, s)$)
  end for
end procedure

3.5.2 Infinite Horizon

3.5.2.1 λ-discounted games

A dynamic programming formulation for one-time policy improvement in the λ-discounted repeated game is

$$\tilde{v}_\lambda(p) = \max_{x,\mu} \min_{y} \Big[\lambda \sum_{k\in K} \sum_{l\in L} \mu^k(l) p^k x^k M^k y + \sum_{m=2}^{\infty} (1-\lambda)^m \big(\sum_{s\in S} \sum_{l\in L} x\bar{\mu}(s,l) u(\tilde{p}^+(p,\mu,x,s,l))\big)\Big] \tag{40}$$

Theorem 4 *One-time policy improvement guarantees a payoff of at least Cav*$[u(p)]$ *for λ-discounted infinite-horizon repeated games.*

Proof: We will prove by construction the existence of a one-time policy improvement strategy that guarantees Cav$[u(p)]$. First note that by applying a basic infinite geometric series result, (40) can be equivalently represented as

$$\tilde{v}_\lambda(p) = \max_{x,\mu} \min_{y} \Big[\lambda \sum_{k\in K} \sum_{l\in L} \mu^k(l) p^k x^k M^k y + (1-\lambda) \sum_{s\in S} \sum_{l\in L} x\bar{\mu}(s,l) u(\tilde{p}^+(p,\mu,x,s,l))\Big]. \tag{41}$$

This is because the expression

$$\sum_{s\in S} \sum_{l\in L} x\bar{\mu}(s,l) u(\tilde{p}^+(p,\mu,x,s,l))$$

remains constant for all $n \geq 2$. Therefore (41) can equivalently be expressed as

$$\max_{x,\mu} \min_{y} \Big[\lambda \sum_{k\in K} \sum_{l\in L} \mu^k(l) p^k x^k M^k y + \sum_{m=2}^{\infty} (1-\lambda)^m \alpha\Big]$$

where

$$\alpha = \sum_{s\in S} \sum_{l\in L} x\bar{\mu}(s,l) u(\tilde{p}^+(p,\mu,x,s,l)).$$

Second, fix $\lambda \in (0,1)$ arbitrarily, and consider a n' such that $\lambda > \frac{1}{n'}$. Set $\hat{\lambda} = \frac{1}{n'}$. Invoking Theorem 6 yields $\tilde{v}_{n'} \geq \text{Cav}[u(p)]$. Third, denote x^* as the optimal stage $m = 1$ mixed action of $\tilde{v}_{n'}$, and similarly denote μ^* as the optimal mixed signal. Since

$$\tilde{v}_{\tilde{\lambda}} \geq \text{Cav}[u(p)]$$

and

$$\sum_{s\in S} \sum_{l\in L} x\bar{\mu}(s,l) u(\tilde{p}^+(p,\mu,x,s,l)) \leq \text{Cav}[u(p)],$$

it follows that the optimal stage $m = 1$ strategy (x^*, μ^*) has the following lower bound

$$\sum_{k\in K} \sum_{l\in L} \mu^{k^*}(l) p^k x^{k^*} M^k y \geq \text{Cav}[u(p)].$$

We then have the following:

$$\lambda \sum_{k\in K} \sum_{l\in L} \mu^{k^*}(l) p^k x^{k^*} M^k y + (1-\lambda) \sum_{s\in S} \sum_{l\in L} x\bar{\mu}(s,l) u(\tilde{p}^+(p,\mu,x,s,l))$$

$$\geq \hat{\lambda} \sum_{k\in K} \sum_{l\in L} \mu^{k^*}(l) p^k x^{k^*} M^k y + (1-\tilde{\lambda}) \sum_{s\in S} \sum_{l\in L} x\bar{\mu}(s,l) u(\tilde{p}^+(p,\mu,x,s,l))\Big]$$

$$\geq \text{Cav}[u(p)].$$

Therefore, by considering mixed action x^* and mixed signal μ^* at stage $m = 1$ and playing non-revealing going forward, player I can guarantee a payoff of $\text{Cav}[u(p)]$ by playing an optimal one-time policy improvement strategy $\tilde{v}_\lambda(p)$.

Corollary 2 *Perpetual policy improvement guarantees a payoff of at least Cav*$[u(p)]$ *for the* λ*-discounted infinite horizon game.*

Proof: By a similar argument used in Corollary 1, we can show that an optimal perpetual-policy improvement strategy guarantees Cav$[u(p)]$.

3.5.2.2 Infinitely repeated games

Theorem 5 *One-time policy improvement is optimal for infinitely repeated games with asymmetric information.*

Proof: We will prove optimality by considering the performance of one-time policy improvement strategies as n grows large. Specifically, we will show that the performance of these strategies converges asymptotically to optimal. Observe that

$$\lim_{n\to\infty} \tilde{v}_n(p) = \max_{x,\mu} \Big[\sum_{s\in S} \sum_{l\in L} x\bar{\mu}(s,l) u(\tilde{p}^+(p,\mu,x,s,l)) \Big] \tag{42}$$

can equivalently be expressed as

$$\max_{p,\alpha} \quad \sum_{l\in L} \alpha_l u(p_l) \tag{43}$$

$$\text{s.t.} \quad \sum_{l\in L} \alpha_l p_l = p$$

by algebraic manipulation. Since the optimal value of (43) is Cav[u(p)], which is also the optimal value of the infinitely repeated games, this concludes the proof.

3.6 LP Formulation

3.6.1 LP Formulation of Incomplete Information Games

Ponssard and Sorin [42] consider the general case of incomplete information games where both players have private information. They showed that finite zero-sum games with incomplete information can be formulated as a linear program (LP) to compute optimal strategies. Let K and R be finite sets. Similar to the asymmetric case, nature chooses $k \in K$ according

to $p \in \Delta(K)$. Since player II also has private information, nature also chooses $r \in R$ according to $q \in \Delta(R)$. Player I and player II can consider pure Bayesian strategies ζ and ψ respectively. Let $\zeta^{(i)}$ be a specific behavior strategy for player I where $i \in I$. Similarly let $\psi^{(j)}$ be a specific behavior strategy for player II. In this section we will denote $x(i)$ to be the probability that pure behavior strategy $\eta^(i)$ is selected. Similarly, we will denote $y(j)$ to be the probability strategy ψ^j is selected. Let $\tilde{M}(\zeta, \psi)$ denote the game payoff to the players for playing the strategy pair (ζ, ψ). The value of the game is V(p,q) and can be computed by solving the following linear program

$$\begin{aligned} \min_{u} \ & \sum_{i=1}^{|I|} u_i \\ \text{s.t. } & \tilde{M}u \geq 1' \\ & u \geq 0 \end{aligned} \tag{44}$$

Let $\beta = (\sum_{i=1}^{|I|} u_i^*)^{-1}$, then $V(p,q) = \beta$ and $x^* = \beta u^*$. Observe that the number of strategies for players I and II are a function of n. Specifically for player

$$|S|^{(K \times \prod_{n=0}^{N} [S \times T]^n)}$$

and for player II

$$|I| = |S|^{(R \times \prod_{n=0}^{N} [S \times T]^n)}.$$

For the asymmetric case, the following lemmas will be used to derive the minimal number of pure strategies necessary for each player to compute optimal policies.

Lemma 3.6.1 *[7] Player I has an optimal strategy in the n-stage and λ-discounted game that depends only, at each m, on m and his beliefs p_m at stage m. In particular, her strategy is independent of the moves of player II.*

Lemma 3.6.2 *[7] Player II has an optimal strategy in the n-stage game and λ-discounted game that depends only, at each m, on m and $(s_1, s_2, \ldots, s_{m-1})$. In particular, his strategy is independent of his own moves.*

Therefore, by the previous lemmas, we have the following. The minimum number of pure strategies necessary to compute optimal strategies for players I and player II are

$$|S|^{(K\times\prod_{n=0}^{N}[S]^n)}$$

and

$$|T|^{(\prod_{n=0}^{N}[S]^n)}$$

respectively.

3.6.2 LP Formulation of Policy Improvement

Theorem 6 *A one-time policy-improvement strategy that guarantees Cav*$[u(p)]$*) can be computed by solving a linear programing problem, and the computational complexity of the linear program is constant with respect to the number of stages of the game.*

Proof: We established in Theorem 3 the existence of a behavior strategy $\tilde{\sigma}^*$ that guarantees a payoff to player I of Cav$[u(p)]$), where $\tilde{\sigma}_1 : k \mapsto \Delta(L) \times \Delta(X)$, $\tilde{\sigma}_{m\geq 2} : \tilde{h_1} \mapsto \Delta(X)$, and $\tilde{h}_1$ is the history at stage $m = 1$ that include the observed signal and actions. *Note that we assume the worst case at stage 1, with respect to signal l, which is that player II can also observe signal l and his strategy at stage* $m = 1$ *can be dependent on the signal.* Recall that if player I uses behavior strategy $\tilde{\sigma}^*$, the best player II can do is also use a strategy $\tilde{\tau}^*$ that has the form $\tilde{\tau}_1 : l \mapsto \Delta(Y)$ and $\tilde{\tau}_{m\geq 2} : \tilde{h}_1 \mapsto \Delta(Y)$. Observe that $\gamma_m^p(\tilde{\sigma}^*, \tilde{\tau}^*) \geq$ Cav[u(p)]) $\forall m \geq 1$ and $\tilde{\sigma}_m^* = \tilde{\sigma}_{m'}^* \; \forall m, m' \geq 2$. Therefore we can express the game payoff as

$$\tilde{\gamma}_\lambda^p(\tilde{\sigma}^*, \tilde{\tau}^*) = (\lambda)\gamma_1^p(\tilde{\sigma}^*, \tilde{\tau}^*) + (1-\lambda)\gamma_2^p(\tilde{\sigma}^*, \tilde{\tau}^*),$$

where $\lambda = \frac{1}{n}$. It is sufficient then to solve for the weighted two-stage game, where $\lambda = \frac{1}{n}$, to compute a strategy for the n-stage game that guarantees Cav$[u(p)]$. For the optimal strategy at stage $m \geq 3$, let $\tilde{\sigma}_m^* = \tilde{\sigma}_2^*$. Since this game has perfect recall (e.g. Past histories are perfectly remembered by each player), we can use Aumann's result on the equivalence

of behavior and mixed strategies. Specifically, the behavior strategy $\tilde{\sigma}^*$ of player I can be equivalently represented as probabilities on pure strategies $\tilde{\zeta}$ where $\tilde{\zeta}_1 : k \mapsto l \times s$, and $\tilde{\zeta}_m \geq 2 : \tilde{h}_1 \mapsto s$ are pure strategies for player I at stage 1 and stage $m \geq 2$ respectively. It follows that by considering a matrix $\tilde{M}$ where element(i, j) denotes the game payoff $\tilde{\gamma}^p_\lambda(\tilde{\zeta}^i, \tilde{\psi}^j)$ for strategy pair $(\tilde{\zeta}^i, \tilde{\psi}^j)$, we can solve for the zero-sum game $\tilde{M}$ using linear programming methods and derive a behavior strategy $\tilde{\sigma}^*$ that guarantees player I Cav$[u(p)]$. Equally important is that the size of the strategy sets for both players are independent of the stages n of the game. Therefore $\tilde{M}$ is invariant with respect to n and so is the computational complexity of the linear program.

Corollary 3 *An optimal strategy for the infinitely repeated game can be computed by solving a LP, and the computational complexity of the LP remains constant with respect to the number of stages of the game.*

Proof: Construct a matrix $\tilde{M}$ as in Theorem 6 with $\lambda = 1$. After solving $\tilde{M}$ using LP methods, an optimal behavior strategy $\tilde{\sigma}$ can then be derived that guarantees an optimal payoff of Cav$[u(p)]$ for infinitely repeated games.

3.6.3 Perpetual Policy Improvement

Perpetual policy improvement consists of repeatedly implementing the one-time policy improvement method at every stage of the game. *Note that this process is essentially receding horizon optimization.* In this section, we will prove that the repeated application of one-time policy improvement also guarantees a payoff of at least Cav$[u(p)]$) in the following theorem.

Theorem 7 *A perpetual policy-improvement strategy that guarantees Cav*$[u(p)]$*) can be computed by solving a linear programing problem online at each stage m, and the computational complexity of the linear program is constant with respect to the number of stages of the game.*

Proof: Construct a matrix $\tilde{M}$ as in Theorem 6. For the n-stage game, set $\lambda = \frac{1}{n}$. For the infinitely repeated game, set $\lambda \approx 1$. Solve the zero sum game $\tilde{M}$ and let the optimal probability distribution of pure strategies ζ be equivalently represented as an optimal behavior strategy $\tilde{\sigma}$. Let x^* and μ^* denote the optimal mixed action and mixed signal respectively derived from $\tilde{\sigma}$ for stage $m = 1$. Consider the signal l and action s realized at stage $m = 1$ and let

$$p_{m=2} = \tilde{p}^+(x^*, \mu^*, s, l).$$

To compute the beliefs at stage $m = 2$, consider the $n-1$ stage game with initial probability $p_{m=2}$. Follow the same process as outlined at stage $m = 1$. Repeat this process for stage m.

3.7 Simulation

3.7.1 Game Setup

As usual, this game consists of two players. We will denote player I as the defender and player II as the attacker. The defender is the row player and maximizer and the attacker is the column player and minimizer. In this example, we assume that the defender knows the type of security system she is defending, and their are two types of security systems (i.e. type I and type II). The probability distribution of the type of security system the defender is defending is uniform (i.e. $p^k = \frac{1}{2}$ for $k = 1, 2$), and there are two stages in this game. Matrix payoffs for the players are

	BR_I	BR_{II}
DC_I	23	375
DC_{II}	−92	69

Type I

	BR_I	BR_{II}
DC_I	−6	−28
DC_{II}	128	−20

Type II

(45)

The defender has the option of choosing a defensive configuration that is the best configuration for the security system she is defending. The defensive configurations will be denoted as DC_I and DC_{II}. Note that the defender has the option of behaving deceptively

by selecting a defensive configuration that is best suited for a different security system. An attacker's actions are to select an attack that is optimized for each possible security system. His actions for this game are BR_I and BR_{II}.

3.7.2 Discussion

We will discuss the performance of four defensive strategies in this section. These strategies are dominant strategy, non-revealing strategy, one-time policy improvement, and perpetual policy improvement. In the one-shot game, the optimal strategy for an defender is to choose the defensive configuration that corresponds to the security system type she is defending. However, for games where $n > 1$, this can be a suboptimal strategy because it can reveal the type of security system to the defender and cost the defender the informational advantage. Specifically, in the two stage game, the defender can achieve a better payoff by selecting the perpetual policy improvement strategy. For games where N is large, the dominant strategy has the worst performance out of the four strategies and the policy improvement strategies have the best performance.

3.7.2.1 Two-stage game

An optimal non-revealing strategy requires the defender, regardless of the type of security, to play as if the system is of type I with probability .70 and to play as if the system is type II with probability .30 at each stage. *A graph of the optimal payoff for a non-revealing strategy with respect to p can be seen in Figure 2.* This strategy rewards the defender with a payoff of 18 and has the worst performance of the four strategies in the two-stage game.[3] One-time policy improvement performs better by guaranteeing an expected payoff of 27. The two strategies differ conceptually only at the first stage as how she plays when using a one-time policy improvement strategy is dependent on type of security system. If the security system is of type I, she selects DC_{II} with probability .92. If the security system

[3]There are games where playing non-revealing is optimal for all n.

is of type II, she selects DC_{II} with probability 1 . At stage two, she selects the configuration DC_{II} with probability .96, which is independent of the actual type of the system. An defender that chooses to use her dominant strategy, which requires her to play honestly, at each stage of the game yields her an expected payoff of 39, which outperforms the two previously mentioned strategies. Perpetual policy improvement yields the defender the highest reward, 53, of the four strategies in consideration. At the first stage of perpetual policy improvement, the defender plays the same way she would have played had she chosen one-time policy improvement. However, the key difference is at the second stage. Instead of playing non-revealing as with the former strategy, the defender selects the defensive configuration that corresponds to the actual system type she is defending. *The optimal payoff for this game is* 63.

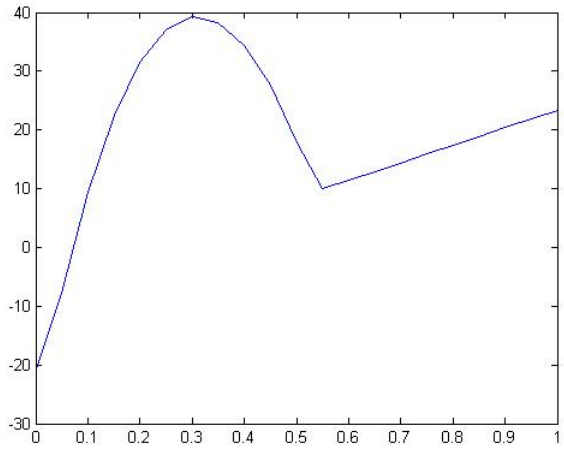


Figure 2: Non-revealing strategy payoff $u(p)$ for security game

3.7.2.2 N-stage game

We discussed the performance of the four strategies in the two-stage case in the previous section and will now examine their performance as N grows large. The expected payoff for the dominant strategy converges asymptotically to 2 and has the worst performance of the four strategies for large N. This is because the attacker can readily learn the type of security system as the defender does not play deceptively. The attacker can then use this

knowledge to select an attack action that is best suited for to the actual type of the system.. An optimal non-revealing strategy performs better than the dominant strategy because the attacker is unable to learn any additional information about the system by observing the defender's action at each stage. As a consequence, the attacker has uncertainty on the most suitable attack action. An defender who chooses this strategy can therefore guarantee a payoff of 18 at every stage. An immediate consequence of this guarantee is that an defender can achieve a game payoff of 18 for games of any length. Policy improvement methods have the best performance of the four strategies for large N. Both methods have identical behavior and converge asymptotically to optimal and yields a payoff of 23. At stage one of the policy improvement methods, the defender behaves deceptively with some probability that dependent on the type of security system. For all stages thereafter, the defender plays a non-revealing strategy that is independent of the security system's type.

3.8 Conclusion

In this chapter, we assume that once the state of the world is selected, the selection remains fixed throughout the duration of the game. Since the players can observe the moves of the other player, each player's knowledge about the past actions of the other changes. This introduces a dynamic aspect to the game as the changing knowledge affects each player's beliefs about the fixed state of the world. Due to the difficulty of computing optimal policies for the informed player, who knows the state, we introduce policy improvement methods based on the ideas of receding horizon control to compute suboptimal strategies. We show that the methods have tight lower bounds and can be computed by solving a linear program whose complexity remains constant with respect to the number of game stages.

Stochastic games with asymmetric information are more general as the state of the world can change during game play. These transitions can be controlled by either player or both. In contrast to repeated games, where the value of the game always exists, the value may not exist in stochastic games when the uninformed player controls the transitions.

However, the key issues and ideas of information exploitation and revelation discussed in the special case, repeated games, are relevant to stochastic games. In particular, computing optimal policies to determine the optimal amount of information to exploit is also prohibitive in stochastic games.

We extend the policy improvement concepts that we applied to repeated games to stochastic games in the next chapter to address the complexity issues in that setting. We were able to incorporate the ideas of non-revelation in our policy improvement methods for the repeated games. However, a key non-revelation result that we exploited is that if the informed player plays non-revealing, the uninformed player's beliefs about the state of the world does not change. In stochastic games, the beliefs can change even when the informed player uses a non-revealing strategy. Therefore, we will explore the concept of minimally revealing strategies and use this strategy as a baseline policy for policy improvement methods for stochastic games. Similar to the repeated games, we are interested in establishing bounds on the performance of these methods.

CHAPTER 4

STOCHASTIC GAMES WITH ASYMMETRIC INFORMATION

4.1 Introduction

In a repeated game, the state of the world is randomly selected by nature and remains fixed throughout the duration of the game. A stochastic game is a repeated game where the state can change from stage to stage according to a transition that is dependent on the current state and the moves of both players. Of interest, in this chapter is the scenario where one of the players has superior information in the stochastic game and also solely controls the state transitions. This scenario creates a complication for the informed player. The complication is that she[1] must decide when and how to use her private information. *Note that by using her private information, she also reveals that information to player II, the uninformed player.* In their seminal work, Aumann and Maschler studied the special case, repeated games with asymmetric information, where every state is absorbing. This work provides insights into the issues that an informed player must address when playing against an opponent that can observe moves and use that information to better estimate the state of the world. Although stochastic games with asymmetric information are more general than the repeated case, the ideas and formulations that Aumann and Maschler introduce extend to the stochastic case. The computational challenges of computing an optimal strategy for the repeated case also extend to the more general stochastic case.

4.1.1 Related Work

Since computing optimal policies for stochastic games with asymmetric information is prohibitive, there is work that considers computing suboptimal policies. Chang and Markus [41] consider a receding horizon approach for two-person zero-sum Markov Games. In their work, the minimizing player selects a "small" horizon and solves the game with the

[1] In this chapter, we will refer to the informed player as "she" and the uninformed player as "he". The assignment of "he" and "she" was arbitrary and was done for the purpose of clarity.

finite horizon (called the subgame) under the assumption that the maximizer makes her decision based on her best performance of the subgame. In [43], Raghavan and Syed consider applying a policy-improvement algorithm to a special class of stochastic games that have additive reward and additive transition structure. They show that for these games, the policy improvement algorithm can generate optimal pure stationary strategies if they exist. In [44] Raghavan explores single-controller stochastic games, where the transition probabilities depend on the actions of the same player in all states. Raghavan shows that non-zero-sum single-controller games can be reduced to linear complementary problems and Lemke's algorithm can be used to find a Nash equilibrium. In [45], Lakshmivarahan investigate conditions under which two learning algorithms playing a zero-sum sequential stochastic game arrives at optimal pure strategies. The algorithms are shown to converge to the optimal pure strategies when they exist with probabilities as close to 1 as desired. In [46], a decentralized learning algorithm is introduced. It is shown that all stable stationary points of the algorithm are Nash equilibrium for the game. For two special cases it is shown that the algorithm always converges to a desirable solution.

4.1.2 Contributions of the Work

We consider single-controller stochastic games with asymmetric information and address the complexity issues for both finite horizon and infinite horizon games by deriving a sub-optimal policy based on the concept of policy improvement. *Policy improvement is a dynamic programming technique that first considers an initial policy, referred to as a baseline policy, that is easily implementable. Through an iterative process, the performance of the baseline policy is compared to that of other policies from a set of candidate policies. If a candidate policy performs better, it becomes the new baseline policy.* The baseline policy we consider is a minimally revealing policy, i.e., one that completes ignores superior information. The improved policy, which is implemented in a receding horizon manner, strategies for the current stage while assuming a minimally-revealing policy for future stages. We derive bounds on the guaranteed performance of the improved policy. Last, we show

that the improved policy can be computed by solving a linear program whose complexity is constant with respect to the game length.

4.1.3 Outline

The outline for the rest of this chapter is as follows. In Section 4.2, we review basic definitions, concepts, and results for stochastic games. In Section 4.3 we introduce single-controller stochastic games with asymmetric information. We consider two classes of these stochastic games in this section, where the player who has superior information also controls the state transitions. In Section 4.4 we present the main results of this chapter. Specifically, we show how policy improvement methods can be applied to single-controller stochastic games with asymmetric information and can provide guarantees on the expected payoff of player I. In Section 4.5 we present receding horizon heuristics and discuss the performance of the heuristics. Last, we present simulations in Section 4.6 that demonstrate the main results of this chapter.

4.2 Preliminaries on Stochastic Games

4.2.1 The Model

A zero-sum stochastic game G is specified by a finite state space Ω, action sets S and J, a map $q : \Omega \times S \times J \mapsto \Delta(\Omega)$, and a reward function $g : \Omega \times S \times J \mapsto \mathbb{R}$. The standard hypothesis that will be assumed is that the play, including the current state, is public. Therefore, the initial state ω_1 and the subsequent states are known by both players. It is assumed that both players can observe the moves of the other and that each player has perfect recall about the previous history h_n. At each stage, both players select a move, $s_n \in S$ for player I, $j_n \in J$ for player 2. The next state ω_{n+1} is selected according to the distribution $q(\cdot \mid s_n, j_n, \omega_n)$ on Ω.

4.2.2 Strategies

For $n = 1, 2, \ldots$, let $H_n = [\Omega \times S \times T]^{n-1} \times \Omega$ be the set of possible histories at stage n. Then $h_n \in H_n$ is a sequence $(\omega_1, s_1, t_1; s_2, t_2; \ldots; \omega_{n-1}, s_{n-1}, t_{n-1}; \omega_n)$ of moves of the two

players in the first $n-1$ stages of the game and the states. Let $X = \Delta(S)$ and $Y = \Delta(T)$ denote the mixed moves of player I and player II respectively where $x_n^k = (x_n^k(s))_{s\in S}$. Let $\zeta_n : h_n \mapsto S$ denote a pure strategy for player I and define $\zeta = (\zeta_1, \zeta_2, \ldots, \zeta_n)$. Similarly let $\psi_n : h_n \mapsto T$ denote a pure strategy of player II and define $\psi = (\psi_1, \psi_2, \ldots, \psi_n)$. Mixed strategies are probability distributions over pure strategies. Behavior strategies are sequences of mappings from H_n to X and mappings from H_n to Y for player I and player II respectively. Aumann [8] showed that mixed strategies can be equivalently represented as behavior strategies for the stochastic games considered in this chapter. Therefore, the terms behavior strategy and mixed strategy will be used interchangeably. Let σ and τ denote the behavior strategies of players I and II respectively. Also let $\sigma(h_n) = (x_n) \in X$ be defined as the vector of mixed moves of player I at stage n.

Definition 4 *A strategy is Markov if it depends only on the stage and on the current state.*

Definition 5 *A strategy is stationary if it depends only on the current state.*

4.2.3 Payoffs

The payoff at stage n is denoted by g_n, where $g_n = g(s_n, j_n, \omega_n)$. Let the game payoff for the n-stage stochastic game be the average of the stage payoffs, denoted by $G_n(\omega)$. Likewise, the payoff for the λ-discounted game and infinite game is denoted by G_λ and G_∞ respectively.

4.2.4 Value of the Game

In [47], Shapley proved that the λ-discounted game has a value and that both players have optimal stationary strategies. Kohlberg [48] then proved that all games with absobing states have a value. Mertens and Neyman [49] introduced the following theorem which states that all stochastic games have a value.

Theorem 8 *[49] For every stochastic game and for every $\epsilon > 0$, there exists a strategy σ of player I and $N > 0$ such that for every n,$n = N, N+1, \ldots, \infty$ and for every strategy τ of*

player II,

$$E_{\sigma,\tau}(\bar{x}_n) \geq v_\infty - \epsilon \tag{46}$$

where $\bar{x}_\infty$ *denotes* $\lim_{n\to\infty} \inf \bar{x}_n$.

As a consequence of the previous theorem, the strategy σ is $\epsilon-$optimal in both the infinite game and in all sufficiently long finite games.

4.2.5 Example: "Big Match"

A game that has played a fundamental role in the development of the stochastic game theory is called "Big Match" and has the following payoff matrix:

	L	R
T	1^*	0^*
B	0	1

Player I has a choice of playing T or B. However, as soon as she plays T the payoff (with a "*") is absorbing. Whatever the payoff receives at the first stage she plays T is the payoff she receives for the remainder of the stages of the game, and therefore, the game is essentially over. If player I plays B, the game is repeated and the stage payoff is given by the entries of the matrix.

The recursive formula v_n for this game is the following:

$$(n+1)v_{n+1} = \text{val}\begin{pmatrix} n+1 & 0 \\ nv_n & 1+nv_n \end{pmatrix} \tag{47}$$

An optimal strategy for player I is to play T with probability $\frac{1}{n+1}$ at the first stage in G_n. The recursive formula for v_λ is the following:

$$v_\lambda = \text{val}\begin{pmatrix} 1 & 0 \\ n = (1-\lambda)v_\lambda & \lambda + (1-\lambda)v_\lambda \end{pmatrix} \tag{48}$$

The optimal payoff for the discounted game is $v_\lambda = \frac{1}{2}$ for all λ. Player I's optimal strategy is to play T with probability $\frac{\lambda}{1+\lambda}$ in G_λ. The optimal strategy for player II is to play $(\frac{1}{2}, \frac{1}{2})$ in both G_n and G_λ.

4.3 Preliminaries on Stochastic games with Asymmetric Information

In this section, we will consider stochastic zero-sum games where player I controls the transitions. Player I controls the transitions if, for every $\omega \in \Omega$ and every $s \in S$, the transition $q(\cdot \mid \omega, s, j)$ does not depend on j. *Player II controls the transitions if the symmetric property holds.* The two classes of games we will consider are the following. The first class is a family of stochastic games where each game has the same Markov chain, but different payoffs. In these stochastic games, player I is informed of the specific stochastic game that is selected by nature. Player II only knows the probability p of the finite family G^k, $k \in K$, of stochastic games on the same state space. In the second class of games, player I is informed of the initial start state ω and the subsequent states during the stochastic game. Player II only knows the transition probabilities q of the Markov chain and the probability distribution p of the initial starting state ω. The expected average payoff up to stage N for these stochastic games with asymmetric information is defined by

$$\gamma_N(p, \omega, \sigma, \tau) = E_{p,\omega,\sigma,\tau}[\bar{g}_N],$$

where $\bar{g}_N = \frac{1}{N}\sum_{n=1}^{N} g^k(\omega_n, i_n, j_n)$ for the first class of games and $\bar{g}_N = \frac{1}{N}\sum_{n=1}^{N} g(\omega_n, i_n, j_n)$ for the second class of games.

4.3.1 Definitions and Concepts

4.3.1.1 Concavification

Consider a continous function $u : \Delta(K) \mapsto \mathbf{R}$, and let $\mathrm{Cav}[u(p)])$ denote the functions concavification. Specifically, let $\mathrm{Cav}[u(p)])$ be the point-wise smallest concave function g on $\Delta(K)$ satisfying $g(p) \geq u(p)\ \forall p \in \Delta(K)$. *In other words, g is the least concave function over $\Delta(K)$.*

4.3.1.2 Communication sets

For $\omega \in \Omega$ denote

$$r_\omega = \min\{n \in \mathbb{N}, \omega_n = \omega\}$$

to represent the first visit to ω.

Definition 6 *Let $\omega_1, \omega_2 \in \Omega$. ω_1 leads to ω_2 if $\omega_1 = \omega_2$ or if $\mathbb{P}_{\omega_1,\sigma}(r_{\omega_2} < +\infty) = 1$ for some strategy σ of player I.*

Definition 7 *$\omega \longleftrightarrow \omega'$ iff ω leads to ω' and ω' leads to ω*

Given $\omega \in \Omega$, denote C_ω to be the communicating set that contains ω and define

$$S_\omega = \{s \in S : q(C_\omega \mid \omega, s) = 1\}$$

Also define

$$\tilde{S}_\omega = \{s \in S : q(\omega \mid \omega, s) = 1\}.$$

Actions in S_ω (respectively $\tilde{S}_\omega$) are called stay actions, and any state ω such that $S_\omega = \emptyset$ is a null state. Denote the set of non-null states as Ω_c.

Lemma 4.3.1 *[9] $\omega \in \Omega_c$ if and only if there is a stationary strategy x_{C_ω} such that C_ω is a recurrent set for x*

Proof: First start with direct implication. Let $\omega \in \Omega_c$. For $\omega' \in C_\omega$, define $x_{\omega'} \in \Delta(A)$ by

$$x_{\omega'} = \begin{cases} 0 & s \notin S_{\omega'} \\ 1/|S_{\omega'}| & s \in S_{\omega'} \end{cases} \tag{49}$$

and let x be any stationary strategy that coincides with $x_{\omega'}$ in each state $\omega' \in C_\omega$.

Lemma 4.3.2 *[9] Assume player I controls the transitions. Let $\omega \in \Omega$ and $\omega' \in C_\omega$. If one of the players can achieve a payoff ϕ in $\Gamma(p, \omega)$, they can also achieve a payoff of ϕ in $\Gamma(p, \omega')$.*

Proof: Assume first that player I can guarantee ϕ in $\Gamma(p, \omega)$. Let σ be a strategy that guarantees $\phi - \epsilon$ in $\Gamma(p, \omega')$, and let σ^* be the strategy that plays x_{C_ω} until r_ω, then switches to σ. In the game $\Gamma(p, \omega')$, the strategy σ^* guarantees $\phi - \epsilon'$ for each $\epsilon' > \epsilon$.

Now assume that player II can guarantee ϕ in $\Gamma(p, \omega)$, but assume to the contrary that he cannot guarantee ϕ in $\Gamma(p, \omega')$ for some $\omega' \in C_\omega$. Since player II cannot guarantee ϕ in ϕ in $\Gamma(p, \omega')$, there is $\epsilon > 0$ such that for every strategy τ of player II and every N there is a strategy $\sigma_{\tau,N}$ of player I and an integer $n_{\tau,N}$ such that $\gamma_{n_\tau,N}(p, \omega', \sigma_\tau, N, \tau) > \phi + \epsilon$. Let $\tau$and N be given. Let σ^* be the strategy of player I defined as follows. Play x_{C_ω} until stage $r_{\omega'}$, then switch to $\sigma_{\tau_\nu,M}$, where τ_ν is the strategy induced by τ after stage ν, and M is sufficiently large so that $\mathbb{P}_{\omega,x_{C_\omega}}(r_{\omega'} < M) > 1 - \frac{\epsilon}{2}$. Since there exists an $n' \geq N$ such that $\gamma_{n_\tau,N}(p, \omega, \sigma^*, \tau) > \phi + \epsilon/2$, there is a contradiction.

4.3.1.3 Minimally revealing strategies

Denote $\hat{\Gamma}_R(p, \omega)$ to be the stochastic zero-sum game with initial state ω, state space C_ω, reward function $\sum_k p^k g^k$, action sets S_ω at each state $\omega' \in C_\omega$ and transition function induced by q. Similarly denote $\tilde{\Gamma}_R(p, \omega)$ to be the n-stage stochastic zero-sum game with initial state ω, state space C_ω, reward function $\sum_k p^k g^k$ and action sets $\tilde{S}_\omega$. Note that the action set consists of stay actions. Therefore, the game play never leaves ω. For the games $\hat{\Gamma}_R(p, \omega)$ and $\tilde{\Gamma}_R(p, \omega)$, if ω is a null state (e.g. the action set is empty), set $\hat{u}(p, \omega) = -\infty$ and $\tilde{u}(p, \omega) = -\infty$ respectively.

4.3.2 Family of Stochastic Games (Level 2)

4.3.2.1 The Model

A zero-sum stochastic game with asymmetric information is described by a finite collection G^k, $k \in K$ of stochastic games on the same state space with a probability distribution $p \in \Delta(K)$. p^k denotes the probability stochastic game G^k is selected. We assume that both players are told the next state ω_{n+1} and the move of the other player. The specific game G^k is chosen by nature from the probability distribution p and the outcome of the selection is told only to player I. We assume that the probability distribution p is public knowledge. The game is then played over several stages. At each stage $n \in N$, the players simultaneously choose an action $s \in S$ and $j \in J$. ω_{n+1} is drawn according to $q(\cdot \mid \omega_n, i_n, j_n)$. Note that since the transition are determined by q^k, player II's beliefs about k can change even if

player I uses a non-revealing strategy (i.e. $x^k = x^{k'}\ \forall\ k, k'$) The game will be parametrized by the initial distribution p and the initial state ω, and will be denoted by $\Gamma(p, \omega)$

Definition 8 *Player I can guarantee $\phi \in R$ in the game $\Gamma(p, \omega)$ if, for every $\epsilon > 0$ there exists a strategy σ of player I and $N \in \mathbf{N}$ such that*

$$\forall \tau, \forall n \geq N,\ \gamma_n(p, \omega, \sigma, \tau) \geq \phi - \epsilon$$

Definition 9 *Player II can defend $\phi \in \mathbf{R}$ in the game $\Gamma(p, \omega)$ if, for every $\epsilon > 0$ and every strategy σ of player* 1*, there exists a strategy τ of player II and $N \in \mathbf{N}$ such that*

$$\forall n \geq N,\ \gamma_n(p, \omega, \sigma, \tau) \leq \phi + \epsilon.$$

4.3.2.2 Optimal strategies

Let the value of the infinite game be denoted by v. Also let $(p, \omega) \in \Delta(K) \times \Omega$ be given. The recursive formula is then

$$v(p_n) = \sum_e \alpha_e \max\left\{\tilde{u}, \max_{\omega' \in C_{\omega_n}, s \notin S_{\omega'}} \mathbf{E}\left[v \mid \omega', s\right]\right\}(\tilde{p}_e, \omega_n), \tag{50}$$

where $\tilde{p}_e \in \Delta(K)$, $\alpha_e \in [0, 1]$, for $e = 1, \ldots, |K| + 1$, such that $\sum_e \alpha_e = 1$ and $\sum_e \alpha_e \tilde{p}_e = p_n$ [9]. An optimal strategy for player I is then the following. If G^k is the game that was selected by nature, player I chooses e according to a state-dependent lottery μ^k, where $\mu^k(e) = \alpha_e \frac{\tilde{p}_e^k}{p_n^k}$. Player I plays a stationary strategy that guarantees $\tilde{u}(p^e, \omega_n)$ in the restricted game $\tilde{\Gamma}_R(p^e, \omega_n)$ if

$$\max\{\tilde{u}, \max_{\omega' \in C_{\omega_n}, s \notin S_{\omega'}} \mathbf{E}\left[v \mid \omega', s\right]\}(\tilde{p}_e, \omega_n) = \tilde{u}(p^e, \omega_n)$$

If

$$\max\{\tilde{u}, \max_{\omega' \in C_{\omega_n}, s \notin S_{\omega'}} \mathbf{E}\left[v \mid \omega', s\right]\}(\tilde{p}_e, \omega_n) = \mathbf{E}\left[v(p^e, \cdot) \mid \omega', s\right]$$

for some $\omega' \in C_{\omega_n}$ and $s \notin S'_{\omega}$, player I plays the stationary strategy $x_{C_{\omega n}}$ until the game reaches the state ω'. At ω', he plays the actions s, and then he recursively switches to a strategy that guarantees $v(p^e, \cdot)$.

4.3.2.3 *Value of the Game*

In repeated games, there exists a value V that can be guaranteed by player I and defended by player II. However, in stochastic games this value does not always exist. Specifically, both the min-max and max-min value can exist but may differ. In [9] it was shown for the case where player I controls the transitions, a value always exists. However, for games where player II controls the state transitions, their are cases where the value does not exist.

4.3.2.4 *Example*

We will now consider examples of stochastic games that were presented and analyzed in [9]. The first game has three states $\Omega = \{\omega_1, \omega_2, \omega_3\}$ and two payoff functions $K = \{1, 2\}$. The states ω_2 and ω_3 are absorbing. Player I can transition from ω_1 to either ω_2 and ω_3 with equal probability by playing the move B. Let $p^{\omega_1} = \frac{1}{8}$, where p^{ω_1} is the probability that the initial starting state is ω_1. The payoff matrices for the stochastic game are

$k = 1$

	L	M	R
T	2/3	2/3	2/3
B	2/3	2/3	2/3

$k = 2$

	L	M	R	
T	2/3	2/3	2/3	ω_1
B	2/3	2/3	2/3	

	L	M	R
T	1	1	1
B	−1	−1	−1

	L	M	R	
T	−1	−1	−1	ω_2
B	1	1	1	

	L	M	R
T	4	0	2
B	4	0	−2

	L	M	R	
T	0	4	−2	ω_3
B	4	0	2	

Let $u_{\omega_i}(p)$ denote the one-stage optimal non-revealing payoff for state ω_i. The non-revealing payoffs in state ω_i are then

$$u_1(p) = 2/3$$

$$u_2(p) = \max\{1-2p, 2p-1\}$$

$$u_3(p) = \begin{cases} 4p & 0 \le p \le 1/4 \\ 2-4p & 1/4 \le p \le 1/2 \\ 4p-2 & 1/2 \le p \le 3/4 \\ 4-4p & 3/4 \le p \le 1 \end{cases}$$

Assume that the game starts in ω_1. If player I chooses a completely revealing strategy that exploits her information, her long-term payoff is $\frac{1}{2}$. This is because the probability that she ends up in state ω_3 is .50 and since player II will learn k, he can ensure that she receives a payoff of no more than 0 in that state. *Note that regardless of what action player II selects in ω_2, player I can guarantee a payoff of* 1. Player I can do better by choosing a non-revealing strategy. If player I plays the move T independent of k for all stages, the state will remain in ω_1 and player can achieve a payoff of $\frac{2}{3}$. A partially revealing strategy yields player I the best long-term payout of $\frac{5}{6}$

In the previous example, player I controls the transitions. If she plays T, she can remain in state ω_1. If she plays B, she will transition to either ω_2 or ω_3 with equal probability. We will now consider an example where player II controls the transitions. As we will see in this example, min-max is not equal to max-min, and therefore this particular game does not have a value. The payoff matrices for this game are

$k = 1$

	j_1	j_2	j_3	j_4	j_5
T	4	0	0	0	0
B	0	0	0	0	0

$k = 2$

	j_1	j_2	j_3	j_4	j_5	
T	0	0	0	0	0	ω_2
B	0	4	4	4	4	

$k = 1$

	j_1	j_2	j_3	j_4	j_5
T	0	1	1	3	0
B	0	1	0	3	0

$k = 2$

	j_1	j_2	j_3	j_4	j_5	
T	3	0	1	0	0	ω_2
B	3	1	1	0	0	

Note that $|\Omega| = |K| = 2$, $|I| = 2$, and $|J| = 5$ for this example. The player II controlled transitions are as follows. If he is in state $]omega_1$ and plays j_5, the game will transition to state ω_2, where ω_2 is absorbing. All other actions in state w_1 are stay actions, where the state does not change.

$$f(p) = u_{\omega_1}(p) = \begin{cases} 3p & 0 \leq p \leq 2 - \sqrt{3} \\ 1 - p(1-p) & 2 - \sqrt{3} \leq p \leq \sqrt{3} - 1 \\ 3(1-p) & \sqrt{3} - 1 \leq p \leq 1 \end{cases} \tag{51}$$

and let

$$g(p) = u_{\omega_2}(p) = 4p(1-p). \tag{52}$$

where $u_{\omega_1}(p)$ and $u_{\omega_1}(p)$ are the optimal non-revealing payoff for the game in states w_1 and w_2 respectively. The max-min value for this game is Cav[min$\{f, g\}$)] when the initial state is w_1, while the min-max value is min$\{\text{Cav}f, g\}$. If we consider the game with $p = 1/2$, the max-min is equal to $3(2 - \sqrt{3})$, while the min-max is equal to $4/5$. Therefore, this game does not have a value when $p = 1/2$.

4.3.3 Unknown Initial State

4.3.3.1 The Model

The game is specified by a state space Ω and a map q. p is a vector that denotes the probability that $\omega' \in \Omega$ is selected as the initial state. The initial state ω' is announced to Player I, while player 2 knows only p. Denote g_n to be the payoff at stage n, where

$$g_n = g(\omega_n, s_n, j_n),$$

ω_n is the current state, and s_n and j_n are the moves of Players I and II respectively at stage n. Let $G_n(p)$ and $G_\lambda(p)$ denote the n-stage and λ-discounted games respectively.

Let $x^\omega \in \Delta(S)$ denote a one-stage strategy. Let A denote the set of public signals and define the conditional probability on Ω given a by:

$$\tilde{p}^\omega(a) = Prob(\omega|a),$$

where

$$Prob(w, a) = \sum_{\omega', s} p^{\omega'} x^{\omega'}(s) q(\omega, a|\omega', s)$$

and

$$Prob(a) = \sum_{w} Prob(\omega, a).$$

A recursive formula for the n-stage game is then

$$v_n(p) = \max_{X^\Omega} \min_{Y} \frac{1}{n} \sum_{\omega, s} p^\omega x^\omega(s) g(\omega, s, j) + \frac{n-1}{n} \mathbf{E}\left[v_{n-1}(\tilde{p}(a))\right]_{p,x}. \tag{53}$$

Note that (66) implies that player I has an optimal Markov strategy in $G_n(p)$ and an optimal Markov stationary in $G_\lambda(p)$, where the state space is $\Delta(\Omega)$. Also note that the recursive formula does not allow player II to recursively construct optimal strategies because computation requires knowledge of x, which is not known to player II, to compute the posterior distribution.

4.4 Policy Improvement (Main Results)

Policy improvement is a dynamic programing approach that consists of first considering an initial policy whose performance is known or can be readily computed. *We will select a minimally-revealing policy as our initial policy in this section and will refer to this policy as the baseline policy.* Next, a candidate policy is selected from some policy set. If the current candidate policy performs better than the baseline policy, the candidate policy becomes the new baseline policy. This new baseline policy is an improvement over the previous baseline

policy, and therefore a policy improvement. This process can then repeated in an iterative manner.

The policy improvement methods presented in this section are based on the ideas of receding horizon optimization. In the n-stage game, player I solves an optimization problem over the interval $[m, n]$, where m is the current stage and n is the total number of stages. We assume that in the future stages, she will only consider minimally revealing strategies. This assumption is key to reducing the complexity of the original dynamic programing formulation $v_n(p, \omega)$ because of the following idea. By playing a minimally revealing strategy in all future stages, a lower bound on her expected future payout over the interval $[m + 1, n]$ can be computed by solving the optimization problem $\tilde{u}(p, \omega)$. Similarly in infinite horizon games, the lower bound on the expected stage payout for all future stages is $\tilde{u}(p, \omega)$, assuming player I plays minimally-revealing over all stages in the future. Note that $\tilde{u}(p, \omega)$ can itself be computed by solving a linear program whose complexity is invariant with respect to n.

Definition 10 *In one-time policy improvement, player I strategizes for the* first stage *of the game while assuming that she will play in a non-revealing manner in all future stages.*

Theorem 9 *The dynamic programing formulation for one-time policy improvement for zero-sum stochastic games where player I controls the transitions is*

$$\tilde{v}_n(p, \omega) = \max_{x,\mu} \min_{y} \Big[\frac{1}{n} \sum_{k,s,j,l} \mu^k(l) p^k x^{k,\omega}(s) y^{\omega}(j) g^k(\omega, s, j) + \frac{n-1}{n} \sum_{s,l,\omega'} \overline{x\mu q}(s, l, \omega') \tilde{u}(\tilde{p}^+(p, \mu, x, s, l, q, \omega, \omega'), \omega') \Big] \tag{54}$$

where $\overline{x\mu q}(s, l, \omega') = \sum_{k \in K} p^k x^k(s) u^k(l) q^k(\omega' \mid \omega, s)$ *and the kth component of* $\tilde{p}^+$ *is*

$$\frac{p^k \mu^k(l) x^k(s) q^k(\omega' \mid \omega, s)}{\sum_{k \in K} p^k x^k(s) \mu^k(l) q^k(\omega' \mid \omega, s)}$$

for the n-stage game.

Theorem 10 *One-time policy improvement guarantees a payoff of* $Cav[\tilde{u}(p,\omega)]$ *for the n-stage game. Equivalently* $\tilde{v}_n(p,\omega) \geq Cav[\tilde{u}(p,\omega)]$

Proof:

First note that if the constraints $x^{k,\omega} = x^{k',\omega}\ \forall k, k'$ and $s \in \tilde{S}_w$ are enforced then

$$\tilde{v}_n(p,\omega) = \max_{x,\mu}\min_{y}\Big[\frac{1}{n}\sum_{k,s,j,l}\mu^k(l)p^k x^{k,\omega}(s)y^{\omega}(j)g^k(\omega,s,j) + \frac{n-1}{n}\sum_{s,l,\omega'}\overline{x\mu q}(s,l,\omega')\tilde{u}(\tilde{p}^+(p,\mu,x,s,l,q,\omega,\omega'),\omega')\Big] \tag{55}$$

$$\geq \max_{x,\mu}\min_{y}\Big[\frac{1}{n}\sum_{k,s,j,l}\mu^k(l)p^k x^{k,\omega}(s)y^{\omega}(j)g^k(\omega,s,j) + \frac{n-1}{n}\sum_{s,l,\omega'}\overline{x\mu q}(s,l,\omega')\tilde{u}(\tilde{p}^+(p,\mu,x,s,l,q,\omega,\omega'),\omega')\Big]$$
$$\text{s.t.}\quad x^k = x^{k'}\ \forall k, k'$$
$$s \in \tilde{S}_\omega \tag{56}$$

$$= \max_{x,\mu}\min_{y}\Big[\frac{1}{n}\sum_{k,s,j,l}\mu^k(l)p^k x^{\omega}(s)y^{\omega}(j)g^k(\omega,s,j) + \frac{n-1}{n}\sum_{s,l}\overline{x\mu}(s,l)\tilde{u}(\tilde{p}^+(p,\mu,x,s,l),\omega)\Big]$$
$$\text{s.t.}\quad x^k = x^{k'}\ \forall k, k'$$
$$s \in \tilde{S}_\omega \tag{57}$$

$$= \max_{x,\mu}\min_{y}\Big[\frac{1}{n}\sum_{l\in L}\overline{\mu}(l)\tilde{u}(\tilde{p}^+(p,\mu,l),\omega) + \frac{n-1}{n}\sum_{l\in L}\overline{\mu}(l)\tilde{u}(\tilde{p}^+(p,\mu,l),\omega)\Big] \tag{58}$$

$$= \max_{x,\mu}\min_{y}\Big[\sum_{l\in L}\overline{\mu}(l)\tilde{u}(\tilde{p}^+(p,\mu,l),\omega)\Big] \tag{59}$$

Through algebraic manipulation (59) can be equivalently represented as

$$= \max_{p,\alpha} \sum_{l \in L} \alpha_l \tilde{u}(p_l, \omega)$$

$$\text{s.t.} \sum_{l \in L} \alpha_l p_l = p \tag{60}$$

Note that the optimal value of (60) is $\text{Cav}[\tilde{u}(p, \omega)]$. Therefore

$$\tilde{v}_n(p, \omega) \geq \text{Cav}[\tilde{u}(p, \omega)].$$

Q.E.D.

Algorithm 2 Perpetual policy improvement

1: **procedure** PERPPOLICYIMPROVE
2: **initialize:** set $p_1 = p$
3: **for** $m = 1 \rightarrow N$ **do**
4: compute $\hat{x}_m$ and $\hat{\mu}$ by solving one-time policy improvement LP with p_m
5: select a move s for type k player using mixed strategy $\hat{x}_m^k$
6: select a signal l for type $k$player using mixed signal $\hat{\mu}$
7: update beliefs vector (i.e. $p_{m+1} = p^+(p, \hat{x}, \hat{\mu}, s, l, \omega)$)
8: **end for**
9: **end procedure**

Definition 11 *In the perpetual policy improvement method, player I strategizes for the first stage while assuming a non-revealing strategy in all future stages.*

Corollary 4 *Perpetual policy improvement guarantees a payoff of Cav*$[\tilde{u}(p, \omega)]$ *for the n-stage game.*

Proof: We will prove by construction. First, consider stage $m = 1$ and implement a one-time policy-improvement strategy. By Theorem 10, there exists a mixed action x^* and a mixed signal μ^* that guarantees a stage payoff of at least $\text{Cav}[\tilde{u}(p, \omega)]$ and a future payoff

of at least Cav$[\tilde{u}(p,\omega)]$. The future payoff can be expressed as $\sum_{l\in L}\alpha^*_{ls}\tilde{u}(p^*_{ls\omega},\omega)$. Next consider stage $m = 2$. Suppose signal l, move s, and state ω were observed at stage 1, then the belief are $p_{ls\omega}$ at stage 2 and the payoff for playing a non-revealing strategy is $u(p_{ls\omega},\omega)$. If player I plays non-revealing over the next $n-1$ stages, she can guarantee an expected payoff of $u(p_{ls\omega},\omega)$. Suppose however that player I implements a one-time policy-improvement strategy at stage $m = 2$. Note a key observation, which is that

$$\tilde{v}_n(p,\omega) \geq \tilde{u}(p,\omega)\ \forall n \geq 1, k \in K.$$

This implies that

$$\tilde{v}_{n-1}(p^*_{ls},\omega) \geq \sum_{m=2}^{n}\frac{1}{n-1}\tilde{u}(p^*_{ls\omega},\omega).$$

Therefore, the worst that can happen to player I if she decides to implement a one-time policy improvement at stage $m = 2$ is that she gets the same payoff $\tilde{u}(p^*_{ls\omega},\omega)$ she would have received if she hadn't deviated from the original policy decided at stage $m = 1$. Consider stage m. By a similar argument, perpetual policy improvement up to stage m guarantees an expected payoff of at least Cav$[\tilde{u}(p,\omega)]$.

Theorem 11 *One-time policy improvement guarantees a payoff of Cav*$[u(p,\omega)]$ *for the* λ*-discounted game. Equivalently* $\tilde{v}_\lambda(p,\omega) \geq Cav[u(p,\omega)]$*, where*

$$\tilde{v}_\lambda(p,\omega) = \max_{x,\mu}\min_{y}\Big[\lambda\sum_{k,s,j,l}\mu^k(l)p^k x^{k,\omega}(s)y^\omega(j)g^k(\omega,s,j) + \sum_{m=2}^{\infty}(1-\lambda)^m\sum_{s,l,\omega'}\overline{x\mu q}(s,l,\omega')u(\tilde{p}^+(p,\mu,x,s,l,q,\omega,\omega'),\omega')\Big] \quad (61)$$

Proof:

First observe that by applying a basic infinite geometric series result, (61) can be equivalently represented as

$$\max_{x,\mu}\min_{y}\Big[\lambda\sum_{k,s,j,l}\mu^k(l)p^k x^{k,\omega}(s)y^{\omega}(j)g^k(\omega,s,j) + (1-\lambda)\sum_{s,l,\omega'}\overline{x\mu q}(s,l,\omega')\tilde{u}(\tilde{p}^+(p,\mu,x,s,l,q,\omega,\omega'),\omega')\Big] \tag{62}$$

because the expression

$$\sum_{s,l,\omega'}\overline{x\mu q}(s,l,\omega')\tilde{u}(\tilde{p}^+(p,\mu,x,s,l,q,\omega,\omega'),\omega')$$

remains constant for all $n \geq 2$.

Note that if the constraints $x^{k,\omega} = x^{k',\omega}\ \forall k,k'$ and $s \in S_w$ are enforced then

$$\max_{x,\mu}\min_{y}\Big[\lambda\sum_{k,s,j,l}\mu^k(l)p^k x^{k,\omega}(s)y^{\omega}(j)g^k(\omega,s,j) + (1-\lambda)\sum_{s,l,\omega'}\overline{x\mu q}(s,l,\omega')\tilde{u}(\tilde{p}^+(p,\mu,x,s,l,q,\omega,\omega'),\omega')\Big]$$

$$\geq \max_{x,\mu}\min_{y}\Big[\lambda\sum_{k,s,j,l}\mu^k(l)p^k x^{k,\omega}(s)y^{\omega}(j)g^k(\omega,s,j) + (1-\lambda)\sum_{s,l,\omega'}\overline{x\mu q}(s,l,\omega')\tilde{u}(\tilde{p}^+(p,\mu,x,s,l,q,\omega,\omega'),\omega')\Big]$$

$$\text{s.t.}\quad x^k = x^{k'}\ \forall k,k'$$

$$s \in S_\omega \tag{63}$$

The constraints imply the following:

$$\sum_{k,s,j,l}\mu^k(l)p^k x^{k,\omega}(s)y^{\omega}(j)g^k(\omega,s,j) = \sum_{l}\bar{\mu}(l)u(p^+(p,\mu,l),\omega)$$

Therefore (63) can be equivalently expressed as

$$\max_{x,\mu}\min_{y}\Big[\lambda\sum_{l}\bar{\mu}(l)u(\tilde{p}^+(p,\mu,x,s,l,q,\omega,\omega'),\omega) + (1-\lambda)\sum_{s,l,\omega'}\overline{x\mu q}(s,l,\omega')u(\tilde{p}^+(p,\mu,x,s,l,q,\omega,\omega'),\omega')\Big]$$

$$= \max_{x,\mu} \min_{y} \Big[\lambda \sum_{l} \bar{\mu}(l) u(\tilde{p}^{+}(p, \mu, x, s, l, q, \omega, \omega'), \omega) + (1 - \lambda) \sum_{s,l,\omega'} \overline{\mu}(l) u(\tilde{p}^{+}(p, \mu, x, s, l, q, \omega, \omega'), \omega') \Big]$$

$$= \max_{x,\mu} \min_{y} \Big[\sum_{l} \bar{\mu}(l) u(\tilde{p}^{+}(p, \mu, x, s, l, q, \omega, \omega'), \omega) \Big] \tag{64}$$

Through algebraic manipulation (64) can be equivalently represented as

$$= \max_{p,\alpha} \sum_{l \in L} \alpha_l \tilde{u}(p_l, \omega)$$

$$\text{s.t.} \quad \sum_{l \in L} \alpha_l p_l = p \tag{65}$$

Q.E.D.

Corollary 5 *Perpetual policy improvement guarantees a payoff of Cav*$[u(p, \omega)]$ *for the* λ*-discounted game.*

Proof: By a similar argument used in Corollary 4, we can show that an optimal perpetual-policy improvement strategy guarantees Cav$[u(p, \omega)]$.

Theorem 12 *A one-time policy-improvement strategy that guarantees Cav*$[\tilde{u}(p, \omega)]$*) can be computed by solving a linear programing problem, and the computational complexity of the linear program is constant with respect to the number of stages of the game.*

Proof: We established in Theorem 10 the existence of a behavior strategy $\tilde{\sigma}^{*}$ that guarantees a payoff to player I of Cav$[u(p)]$), where $\tilde{\sigma}_1 : k \times \omega \mapsto \Delta(L) \times \Delta(X)$, $\tilde{\sigma}_{m \geq 2} : \tilde{h_1} \times \omega \mapsto \Delta(X)$, and $\tilde{h}_1$ is the history at stage $m = 1$ that include the observed signal and actions. *Note that we assume the worst case at stage 1, with respect to signal l, which is that player II can also observe signal l and his strategy at stage* $m = 1$ *can be dependent on the signal.* Recall

that if player I uses behavior strategy $\tilde{\sigma}^*$, the best player II can do is also use a strategy $\tilde{\tau}^*$ that has the form $\tilde{\tau}_1 : l \mapsto \Delta(Y)$ and $\tilde{\tau}_{m\geq 2} : \tilde{h}_1 \times \omega \mapsto \Delta(Y)$. Observe that $\gamma_m^p(\tilde{\sigma}^*, \tilde{\tau}^*) \geq \text{Cav}[\tilde{u}(p)])\ \forall m \geq 1$ and $\tilde{\sigma}_m^* = \tilde{\sigma}_{m'}^*\ \forall m, m' \geq 2$. Therefore we can express the game payoff as

$$\tilde{\gamma}_\lambda^p(\tilde{\sigma}^*, \tilde{\tau}^*) = (\lambda)\gamma_1^p(\tilde{\sigma}^*, \tilde{\tau}^*) + (1-\lambda)\gamma_2^p(\tilde{\sigma}^*, \tilde{\tau}^*),$$

where $\lambda = \frac{1}{n}$. It is sufficient then to solve for the weighted two-stage game, where $\lambda = \frac{1}{n}$, to compute a strategy for the n-stage game that guarantees $\text{Cav}[\tilde{u}(p)]$. For the optimal strategy at stage $m \geq 3$, let $\tilde{\sigma}_m^* = \tilde{\sigma}_2^*$. Since this game has perfect recall (e.g. Past histories are perfectly remembered by each player), we can use Aumann's result on the equivalence of behavior and mixed strategies. Specifically, the behavior strategy $\tilde{\sigma}^*$ of player I can be equivalently represented as probabilities on pure strategies $\tilde{\zeta}$ where $\tilde{\zeta}_1 : k \times \omega \mapsto l \times s$, and $\tilde{\zeta}_m \geq 2 : \tilde{h}_1 \times \omega \mapsto s$ are pure strategies for player I at stage 1 and stage $m \geq 2$ respectively. It follows that by considering a matrix $\tilde{M}$ where element(i, j) denotes the game payoff $\tilde{\gamma}_\lambda^p(\tilde{\zeta}^i, \tilde{\psi}^j)$ for strategy pair $(\tilde{\zeta}^i, \tilde{\psi}^j)$, we can solve for the zero-sum game $\tilde{M}$ using linear programming methods and derive a behavior strategy $\tilde{\sigma}^*$ that guarantees player I $\text{Cav}[\tilde{u}(p)]$. Equally important is that the size of the strategy sets for both players are independent of the stages n of the game. Therefore $\tilde{M}$ is invariant with respect to n and so is the computational complexity of the linear program.

Corollary 6 *An optimal strategy for the infinitely repeated game can be computed by solving a LP.*

Proof: Construct a matrix $\tilde{M}$ as in Theorem 12 with $\lambda = 1$. After solving $\tilde{M}$ using LP methods, an optimal behavior strategy $\tilde{\sigma}$ can then be derived that guarantees an optimal payoff of $\text{Cav}[u(p)]$ for infinitely repeated games.

4.5 Heuristic Receding Horizon Policies

In this section, we consider applying receding horizon optimization methods to the class of stochastic games where player II has uncertainty about the current state of the world ω.

Note that these games were discussed in Section 4.3.3. Recall that the recursive formulation for the value of the game is

$$v_n(p) = \max_{X^\Omega} \min_{Y} \frac{1}{n} \sum_{\omega,s} p^\omega x^\omega(s) g(\omega, s, j) + \frac{n-1}{n} \mathbf{E}\left[v_{n-1}(\tilde{p}(a))\right]_{p,x}. \tag{66}$$

Similar to the repeated games, the complexity of evaluating the Bellman's equation can be attributed to the recursive nature of the cost-to-go function $\mathbf{E}\left[v_{n-1}(\tilde{p}(a))\right]_{p,x}$. By assuming minimally revealing policies for the informed player, we were able to achieve an exact, non-recursive, and non-trivial function to compute the cost-to-go for repeated games and a non-trivial lower bound estimate for the level 2 class of stochastic games discussed in Section 4.3.2.

Unfortunately the technique of assuming minimally revealing policies does not provide us with a reasonable and non-trivial cost-to-go function for the class of games considered in section. However, we can still apply the ideas of receding horizon optimization to this problem. We can make the following assumption. We will assume that the payoff player I receives for playing a minimally revealing policy from stage $m+1$, she also receives that payoffs for stages $m+2$, $m+3$, $\ldots N$. This assumption does not hold generally because player II's beliefs p_m at stage m can change significantly even if player I plays minimally revealing. Consider the stochastic game illustrated in Figure 3 as an example of such a scenario. Observe that for any distribution p of initial states, the beliefs of player II from stages $m \geq 2$ onward will be $p(\omega_2) = 1$. This change in beliefs of player II occurs because of the transition probability q for the game. No matter where the game starts at stage $m = 1$, the game will transition and remain in state ω_2 in all future stages. Therefore, player II's beliefs can change significantly even when player I uses a minimally revealing strategy.

4.5.1 Receding Horizon Formulation

A formulation for a 1-step receding horizon formulation is the following:

$$\tilde{v}_n(p) = \max_{X^\Omega} \min_{Y} \Big[\frac{1}{n} \sum_{\omega,s} p^\omega x^\omega(s) g(\omega, s, j) + \frac{n-1}{n} \mathbf{E}\left[\tilde{u}(\tilde{p}(a))\right]_{p,x}\Big]. \tag{67}$$

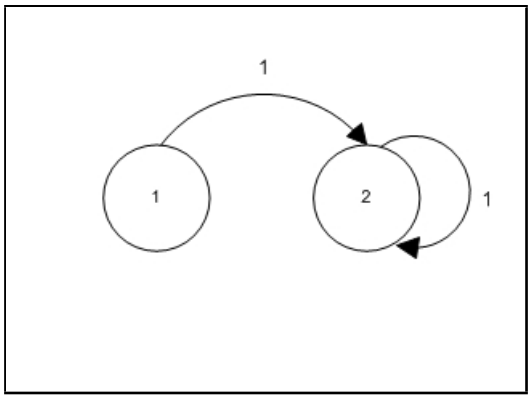


Figure 3: Stochastic game where player II's beliefs can change significantly even when player I uses a minimally revealing strategy

Observe that the key difference between equations (66) and (67) is the cost-to-go function, where the cost-to-go function is $\mathbf{E}\left[\tilde{u}(\tilde{p}(a))\right]_{p,x}$ for the receding horizon formulation. Note that we can also consider a m-step receding horizon formulation. Similar to the 1-step formulation, the m-step formulation would approximate the future payoffs as the average of the next m stages where we assume that a minimally revealing strategy is played during those stages. Although an m-step formulation can be more favorable than the 1-step formulation and offers significant computational savings with respect to the computation of optimal policies, the computational complexity still grows substantially with each additional "look-ahead step" in the future.

4.5.2 LP Formulation of Receding Horizon Optimization

The m-step receding horizon optimization problem (i.e. equation (67)) can be formulated as a linear program. We will consider the case where $m = 1$. However, a similar process can be followed for arbitrary m. The formulation is similar to that of Theorem 12. The strategies for player I are $\tilde{\sigma}_1 : \omega \mapsto \Delta(X)$ and $\tilde{\sigma}_{m\geq 2} : \tilde{h_1} \times \omega \mapsto \Delta(X)$, and the strategy for player II is $\tilde{\tau}_m : \tilde{h}_1 \mapsto \Delta(Y)$.

4.6 Simulation

4.6.1 Game Setup

The following example demonstrates how policy improvement can be applied to stochastic games with asymmetric information to compute suboptimal solutions that have guarantees. As usual, player I is the row player and maximizer, and player II is the column player and minimizer. In policy improvement, player I's objective is to strategic for the current stage of the stochastic game, while assuming a non-revealing payoff in future stages. We will consider the n-stage stochastic game where the game payoff is an average of the stage payoffs. The game has two states $\Omega = \{\omega_1, \omega_2\}$. there are two possible payoff functions (e.g. $K = 1, 2$). The probability distribution over K is initially assumed to be uniform.

Payoff structure for the game is the following:

$k = 1$

	L	C	R	ω^+
T	4	0	2	–
M	4	0	−2	–
B	1/3	1/3	1/3	ω_2

$k = 2$

	L	C	R	ω^+
T	0	4	−2	–
M	0	4	2	–
B	1/3	1/3	1/3	ω_2

ω_1

	L	C	R	ω^+
T	5	5	0	–
M	1/4	1/4	1/3	–
B	0	0	0	–

	L	C	R	ω^+
T	0	0	0	–
M	1/3	1/4	1/4	–
B	0	5	5	–

ω_2

In this game, player I has two stay actions in ω_1 , which are T and B. *Recall that playing a stay action keeps the game in the current state with probability* 1. Selecting action M will transition the game to state ω_2 with probability 1. In state ω_2, all actions are stay actions and ω_2. Therefore, ω_2 is said to be absorbing.

4.6.2 Discussion

We will discuss the performance of four possible strategies in this section. These strategies are dominant strategy, non-revealing strategy, one-time policy improvement, and perpetual policy improvement. We will first consider the case where $N = 2$ and then proceed to discuss the performance of these strategies as N grows large. We will assume that the game starts at state ω_1.

4.6.2.1 Two-stage game

A strategy that player I can consider is to play an optimal non-revealing strategy. This strategy consists of player I selecting action B at stage $m = 1$ for a stage payoff of 1/3 and selecting actions T and B with equal probability for stage $m = 2$ that has a stage payoff of 1.25. The expected game payoff for non-revealing is .7917. Player I can achieve a

game payoff of 1 for playing her dominant strategy. Note that in this context, the dominant strategy is a greedy strategy. In state ω_1, she will play T if $k = 1$ and B if $k = 2$ and will achieve a stage payoff of 2. With this strategy the game remains in state ω_1 at stage 2 and since the dominant strategy if fully revealing, player II will know whether the payoffs are of type 1 or 2 and can ensure that player I achieves a stage payoff of 0 and a game payoff of 1. Policy improvement also guarantees a game payoff of 1. At stage 1, player I plays T with probability .75 if k=1 and B with probability .75 if k=2. At stages 2 she plays non-revealing, which guarantees a stage payoff of 1.

4.6.2.2 N-stage game

The dominant strategy has the worst performance and converges asymptotically to 0 with respect to the number of stages in the game. The non-revealing strategy performs better by guaranteeing a game payoff of .7917. Policy improvement performs the best by guaranteeing a game payoff of 1.

CHAPTER 5

CYBER ATTACK FORECAST MODELING USING A GAME-THEORETIC FRAMEWORK

5.1 Introduction

A reactive mindset is often the status-quo in the security community. Consider popular security products that include intrusion detection systems (i.e. Snort) and anti-virus software as examples.[1] In the case of anti-virus (AV) software, the typical scenario is the following. First, new malware is developed by cyber hackers and then tested against popular versions of AV software tools.[2] *Testing malware against AV tools virtually guarantees that new malware will initially go undetected.* Next, the malware infects computing devices, and after some period of time has elapsed, which can range from days to years, AV signatures are developed by the AV vendors (i.e Symantec, Kaspersky, and McAfee).[3] This cat-and-mouse pattern then repeats itself and can be observed within and across the security community.

Reactive security methodologies are effective against novice adversaries because these adversaries typically use off-the-shelf tools and implement popular hacking techniques. In contrast, the pioneering adversaries are able to push both cyber-hacking and security frontiers forward by developing new malware, designing advanced hacking techniques and methodologies, and challenging security researchers and practitioners to match their ingenuity.[4] Measures and approaches that are forward looking and predictive are needed to combat these adept adversaries and to address advanced persistent threats (APTs). Proactive security is an approach that has the potential to address the problems that adept and well funded adversaries present.

[1] A consequence of reactive security tools is that once a threat has been detected, damage has been done and costs have been incurred.

[2] Websites such as `http://www.virustotal.com` can test malware against over 44 different AV tools. Although these sites are intended for security professionals, they can also be used by hackers.

[3] Flame malware that was discovered in May 2012 had been operating in the wild since February 2010.

[4] Malware developers have introduced polymorphic viruses that mutate the machine code of the virus after each execution. This polymorphism feature is designed to defeat AV tools that look for patterns in new viruses that match older versions of the viruses.

Actionable cyber-attack forecasting is a proactive approach that is considered in this thesis. The objectives of actionable cyber-attack forecasting are to learn an attacker's behavioral model, to predict future attacks, and to select appropriate countermeasures to prevent future attacks. Impediments that have prevented the realization of reliable cyber-attack forecasting include, but are not limited to, difficulty in modeling the adversary in an analytical framework and the computational complexity of analyzing the model to forecast attacks. Computational complexity issues will be addressed in this chapter.

5.2 Related Work

Previous work to address forecasting challenges, where uncertainty exists about the capabilities of an attacker includes the work of Alpcan et al. and You et al. In [50], Alpcan et al. model the interaction between attackers and an Intrusion Detection System (IDS) using a stochastic (Markov) game. The defender operates the IDS and has uncertainty about the attacker's intent. Tools that include value iteration are used to solve Markov Decision Processes. In [51], You et al. describe how to model cyber-security problems that consider the interaction between an attacker and a defender in a two-player zero-sum game. They illustrate how the Nash and Bayesian Equilibria can be used to predict the behavior of an attacker and to analyze the interaction between attacker and defender. You et al. suggest that linear programs could be used to solve these problems.

Similar to You et al., cyber-security problems are also modeled as Bayesian zero-sum games in this chapter. Computational methods are introduced to approximate solution to cyber-security problems. A key feature of these methods is that the solution can be computed by solving a linear program whose complexity is invariant with respect to the number of stages of the zero-sum game. These computational methods also have tight lower bounds on their performance that converge asymptotically to optimal with respect to the number of stages of the game.

5.3 Outline

The outline of the remainder of this chapter is as follows. In Section 5.4, *iCTF2010*[5], a cyber-security challenge problem, will be discussed. In Section 5.5, asymmetric information games[6] will be introduced and basic concepts and definitions will then be discussed. In Section 5.6, adversarial models for capture-the-flag (CTF) will be developed and techniques to reduce the complexity of the models will be explored. The CTF problem will then be formulated as a security game with asymmetric information in Section 5.7. Last, simulations will be used to demonstrate how the models and techniques developed in this chapter can be used to learn the behavioral model of an adversary, to predict future attacks, and to launch appropriate countermeasures.

5.4 iCTF2010

5.4.1 Overview

Security researchers at the University of California Santa Barbara (UCSB) host a live capture-the-flag tournament each year [52]. *The 2010 version of this tournament, called iCTF2010, will be considered in this chapter.* There are typically over 900 participants in the tournament, and the participants are hackers from the international community. The purpose of the tournament is to observe and analyze strategies and techniques of real hackers and collect datasets that can be used in security research projects. *iCTF2010* is designed as an abstraction of real-word cyber-security scenarios. For instance, consider the security system at Georgia Tech as a target. The various departments of Georgia Tech (i.e. ECE, ME, etc.) are subsystems that are required to run certain services such as ssh, smtp, citrix, to facilitate the computing needs of faculty, staff and students.[7] The attacker's objective is

[5]This challenge problem will be modeled as a strategic game.

[6]This game theoretic framework will be used later in the chapter to model the capture-the-flag (CTF) problem as a security game.

[7]An operational state of each subsystem in this example can be defined as the services that are not offline because of maintenance. Maintenance could consist of security patch updates and/or server upgrades that effect a particular service.

to successfully disrupt critical services based on partial information he[8] receives about the operational state of the system.[9]

The CTF challenge problem is abstracted as a controlled partially-observable stochastic system, and each subsystem is modeled as a Markov chain. A complication for the attacker is that he does not have full knowledge of the operational state. However, he can estimate the state based on a subset of emitted signals that are correlated with the state transitions. Given those estimates, he can then chooses an appropriate action that corresponds to the critical services he desires to disrupt. The objective of the attacker is to cause maximal disruption to the overall system. In the proceeding section, a formal description of *iCTF2010* will be presented. The original CTF challenge problem was modified in this chapter, and these modifications will be discussed. *The original CTF formulation captures aspects of real-world cyber-security problems that include a dynamic security system whose behavior is at least partially observable and an adversary that can potentially learn and predict the system's behavior.* The modifications presented in this chapter incorporate an additional characteristic of real-world cyber-security problems. This characteristic is a defender who initially has some uncertainy about the capabilities and behavior of an attacker, but who can potentially learn an attacker's capabilities and predict his behavior by using previous observations.

5.4.2 Model description

5.4.2.1 Target System

The target system is abstracted as a discrete-time finite state, finite output Hidden Markov Model (HMM). The target system will be referred to as T composed of N subsystems $T_{(1)}, T_{(2)}, ..., T_{(N)}$. The operational states of subsystem T_i are denoted by

$$\mathbb{A}_{(i)} = \{a_{(i)_1}, a_{(i)_2} \ldots, a_{(i)_{n_i}}\}.$$

[8]The attacker will be referred to as "he," and the defender will be referred to as "she." The assignment of "he" and "she" was arbitrary and was done for the purpose of clarity.

[9]It will be assumed that each subsystem hosts its own services and is therefore not impacted by service disruptions in other subsystems.

The state space of the target system is

$$\mathbb{A} = \mathbb{A}_{(1)} \times \mathbb{A}_{(2)} \times \ldots \times \mathbb{A}_{(N)}.$$

The set of observation signals of subsystem T_i is denoted by

$$\mathbb{B}_{(i)} = \{b_{(i)_1}, b_{(i)_2} \ldots, b_{(i)_{m_i}}\}.$$

The observation signal of the attacker at every instant is an unordered N-tuple of output symbols generated by the subsystems. Given the observation signal $\bar{B} = \{\bar{b}_1, \ldots, \bar{b}_N\}$ at some instant, the attacker knows that there exists an ordering of the elements of the signal, say $(\bar{b}_{\sigma(1)}, \ldots, \bar{b}_{\sigma(N)})$, where $\sigma : \{1, \ldots, N\} \to \{1, \ldots, N\}$ is a bijection, such that

$$(\bar{b}_{\sigma(1)}, \ldots, \bar{b}_{\sigma(N)}) \in \mathbb{B}_{(1)} \times \mathbb{B}_{(2)} \times \ldots \times \mathbb{B}_{(N)}.$$

He may not know the particular ordering, but he can always perform some probabilistic inference given the HMM abstraction. The output space of the target system is denoted by $\mathbb{B}$. After the states and outputs have been relabeled, they are denoted by $\mathbb{A} = \{a_1, \ldots, a_n\}$ and $\mathbb{B} = \{b_1, \ldots, b_m\}$, where $n = \prod_{i=1}^{N} n_i$ and m is the number of unordered N-tuples of output symbols generated by the subsystems.

It is assumed that the state transitions of each subsystem are independent from each other. Let $\{(X_t, Y_t)\}_{t \in \mathbb{Z}_+}$ denote the state and output process of the given HMM. The statistical description of the model is then given by an initial distribution vector π, where

$$\pi_i = \mathbf{Pr}[X_t = a_i], \quad i \in \{1, \ldots, n\},$$

and a set of m transition matrices $\{M[y_1], \ldots, M[y_m]\}$ where

$$M[y_k]_{ij} = \mathbf{Pr}[Y_{t+1} = b_k, X_{t+1} = a_i \mid a_t = x_j],$$

where $k \in \{1, \ldots, m\}, i, j \in \{1, \ldots, n\}$.

5.4.2.2 *Attacker*

The action set of the attacker is denoted by

$$\mathbb{S} = \mathbb{S}_{(1)} \times \ldots \times \mathbb{S}_{(N)}$$

and corresponds to the set of services he disrupts in each subsystem. A payoff structure $r_{\mu,\lambda} : \mathbb{X} \times \mathbb{S} \rightarrow \mathbb{R}$ is associated with the action pairs of the state. This payoff structure reflects whether an attacker, based on his information, chose to disrupt a service that is relevant to the current operational state of the system. The subscript μ reflects the skill level of the attacker and λ is associated with the resources that the defender allocates to the system. The notion of probing[10] is relevant to the attacker's problem. The payoff incurred at each time step provides the attacker with additional information that can be used to estimate the current operational state. Therefore, the attacker is faced with the problem of leveraging short term payoff versus obtaining more accurate information about the current state that will prove beneficial in the long run.

5.4.2.3 *Defender*

The defender's objective in the challenge problem is to minimize the cost of an attack by an adversary on the target system. Since an attacker's goal is to target critical services, the defender can allocate resources to protect these critical services from being disrupted. Let λ_j represent the amount of resources that the defender allocates to protecting service s_j. The likelihood of a successful attack on s_j decreases as λ_j increases. Deciding how to allocate resources among the services can be challenging for the defender because of her uncertainty about an attacker's type. This resource allocation issue is discussed in Section 5.7.2.2.

[10]The idea of optimal probing is an interesting topic that will be considered in future research.

5.5 Asymmetric Information Games

5.5.1 Overview

In a repeated zero-sum game, two players (defender and attacker) repeatedly play the same zero-sum game over several stages. It is assumed that while both players can observe the actions of the other, only the attacker knows the specific opponent he is playing against. Although the defender has uncertainty about the type of attacker she faces, she has a probability distribution of attacker types and can use her observation of the attacker's actions during game play to eventually learn the attacker's type. The dilemma faced by an attacker is how should he trade off the short-term reward by exploiting his private information versus the long-term consequences resulting from revelation of his type.

Classic work by Aumann and Maschler [10] derives a recursive formula for the value of the game, which quantifies the exploitation tradeoff, and also derives the optimal policy for the attacker. Aumann and Maschlers' model for explicit computations of optimal policies is prohibitive for games with multiple stages. In [53], this computational issue is addressed by introducing methods to compute suboptimal strategies by solving linear programs whose complexity is constant with respect to the number of stages. The methods from [53] are discussed in Section 5.7.2.

5.5.2 Game Setup

5.5.2.1 Game Play

Two players repeatedly play a zero-sum matrix game over N stages. The attacker is the row player and maximizer, and the defender is the column player and minimizer. There are a finite set K of possible attacker types that the defender can face. A specific attacker is chosen from this set to play against the defender. Let S be the set of pure strategies of an attacker, and similarly define J to be the set of pure strategies of the defender. The payoff matrix for an attacker of type k will be denoted as $M^k \in \mathbb{R}^{|S|\times|J|}$. Before the initial stage $m = 1$, nature selects an attacker type according to a probability distribution $p \in \Delta(K)$, which is common knowledge. The outcome of this selection is not revealed to the defender.

Once selected, the attacker's type remains fixed over all stages of the game.

5.5.2.2 Strategies

Mixed strategies correspond to distributions over pure strategies. Let $x_m^k \in \Delta(S)$ denote the mixed strategy of an attacker of type k at stage m. In repeated play, this strategy can be a function of the actions of both players during stages $1, ..., N$. Likewise, let $y_m \in \Delta(J)$ denote the mixed strategy of the defender at stage m, which again can depend on player actions over stages m=1,...,N. Let $x_m = \left\{x_m^1, ..., x_m^K\right\}$ denote the collection of mixed strategies for all attacker types and for all states at stage m, and $x = \{x_1, ..., x_N\}$ denote mixed strategies over all states and stages. Likewise, let $y = \{y_1, .., y_N\}$ denote the defender's mixed strategies over all stages.

5.5.2.3 Payoffs

Let

$$\gamma_m^p(x, y) = \sum_{k \in K} p^k x_m^k M^k y_m$$

denote the expected payoff for the pair of mixed strategies (x, y) at stage m. The payoff for the n-stage game is then defined as

$$\bar{\gamma}_n^p(x, y) = \frac{1}{n} \sum_{m=1}^{n} \gamma_m^p(x, y). \tag{68}$$

5.5.3 Concepts and Definitions

5.5.3.1 Beliefs

Since the defender is not informed of the attacker's type k, she can build beliefs on the type. These beliefs are a function of the initial distribution p of attacker types and the observed moves of an attacker. An attacker must therefore carefully consider his actions at each stage as they could potentially reveal his type to the defender. To get a worse case estimate of how much information an attacker transmits about his type through his actions, he models the defender as a Bayesian player and assumes that the defender knows his mixed strategy.

The updated belief p^+ is computed as

$$p^+(p, x, s) = \frac{p^k x^k(s)}{\bar{x}(p, s)} \tag{69}$$

where $\bar{x}(p, s) := \sum_{k \in K} p^k x^k(s)$ and $x^k(s)$ is the probability that an attacker of type k plays pure action s.

5.5.3.2 Non-revealing strategies

Revealing information is defined as an attacker selecting a mixed strategy that is dependent on his type k. From (69), it follows that a mixed strategy x_m at stage m does not change the current beliefs of the defender if $x_m^k = x_m^{k'} \; \forall \; k, k'$. As a consequences, an attacker who plays as if he is oblivious of his type, ensures that his opponents beliefs about his type do not change.[11]

An optimal non-revealing strategy can be computed by solving

$$u(p) = \max_{x \in \mathrm{NR}} \min_{y} \sum p^k x^k M^k y, \tag{70}$$

where

$$\mathrm{NR} = \{x_m \,|\, x_m^k = x_m^{k'} \;\; \forall k, k' \in K\}$$

is the set of non-revealing strategies [6]. By playing an optimal non-revealing strategy at each stage of the game, an attacker can guarantee a game payoff of $u(p)$.[12]

Definition 12 *Let Cav*$[u(p)]$ *denote the point-wise smallest concave function* g *on* $\Delta(K)$ *satisfying* $g(p) \geq u(p) \; \forall p \in \Delta(K)$.

5.5.3.3 Short-term vs. long-term payoffs

The dynamic programming recursive formula

$$v_{n+1}(p) = \max_{x_1} \min_{y_1} \left[\frac{1}{n+1} \sum_{k \in K} p^k x_1^k M^k y_1 + n \sum_{s \in S} \bar{x}_s v_n\Big(p^+(p, x_1, s)\Big) \right], \tag{71}$$

[11] In stochastic games, it is possible for the defender's beliefs about an attacker's type to change even if an attacker plays as if it is oblivious of its type.

[12] In [53], this idea of non-revelation was exploited to reduce the complexity of Aumann and Maschler's formulation.

introduced by Aumann and Maschler [10], characterizes the value of repeated zero-sum games with asymmetric information. Note that n is a non-negative integer. When $n = 0$, the problem reduces to

$$v_1(p) = \max_{x_1} \min_{y_1} \sum_{k \in K} p^k x_1^k M^k y_1, \tag{72}$$

which is the value of the one-shot zero-sum game.

A key interpretation of this formulation is that it also serves as a model of the tradeoff between short-term gains and the long-term informational advantage. For each decision x_1 of an attacker, the model evaluates the payoff for the current stage, which is represented by the expression $\sum_{k \in K} p^k x_1^k M^k y_1$, and the long-term cost for decision x_1, which is represented by $n \sum_{s \in S} \bar{x}_s v_n\Big(p^+(p, x_1, s)\Big)$.

It is worth pointing out that the computational complexity of finding the optimal decision x_1 can be attributed to the cost of calculating the long-term payoff. Since the long-term payoff is a recursive optimization problem that grows with respect to the game length, it can be difficult to find optimal strategies for games of arbitrary length. This difficulty is because the number of decision variables in the recursive optimization problem grows exponentially with respect to the game length. A revised formulation

$$\hat{v}_n(p) = \max_{x_1} \min_{y_1} \left[\frac{1}{n} \sum p^k x_1^k M^k y_1 + (n-1) \sum_{s \in S} \bar{x}_s u\Big(p^+(p, x_1, s)\Big) \right] \tag{73}$$

was introduced in [53] to address a complexity issue of the recursive formulation of the value of the game. In this formulation, it is assumed that the informed player uses optimal non-revealing strategies (i.e. $u(p)$) for all future stages. Therefore, the cost-to-go function $v_n(p)$ in (71) can be expressed as $u(p)$ in (73). As a consequence of the non-revealing assumption, the computational complexity remains constant with respect to the number of stages of the game.

Theorem 13 *[53] A perpetual policy improvement strategy can be computed by solving a linear program online at each stage of the game, and the computational complexity of the linear program is constant with respect to the number of stages of the game.*

In [53], lower bounds on $\hat{v}_n(p)$ were established. It was also shown that the lower bounds were tight, and it was proved that $\hat{v}_n(p)$ has asymptotic convergence to optimality with respect to the number of stages n.

Theorem 14 *[53] One-time policy improvement and perpetual policy improvement achieve Cav[u(p)] and the optimality bounds are*

$$Cav[u(p)] \leq \hat{v}_n(p) \leq v_n(p) \leq Cav[u(p)] + \frac{C}{\sqrt{n}} \sum_{k \in K} \sqrt{p^k(1-p^k)} \tag{74}$$

The computational complexity of the attacker models will be reduced by using the $\hat{v}_n(p)$ formulation described in Section 5.5.3.3.

5.6 Attacker Modeling and Complexity Reduction

A basic adversarial model should addresses the following questions about a specific adversary:

1. What are its skills? **(Skillset/Capabilites)**

   Since each critical service can require specific technical skills to be disrupted, what is the probability that an attacker can successfully disrupt service s_j?

2. What is its intent? **(Intent)**

   Is the ultimate goal of an adversary to prevent the success of the security system under consideration or just to create general disruption?

3. How patient is the adversary? **(Patience)**

   Is the attacker greedy, or is it patient and willing to forgo an immediate gain to maximize its long-term payoff?

4. How does it build beliefs about the system's current state? **(Beliefs)**

   Is computing the system's belief function computationally prohibitive? If so, what technique does the adversary use to approximate the belief function?

5. How does it make decisions based on its state estimates? **(Strategies)**

   Given an estimate of the system, will an adversary disrupt critical services of the most likely operational state or disrupt services that maximize its expected payoff?

In the adversarial models developed in this section, assumptions will be made that allow the main ideas of attacker modeling and complexity reduction techniques to be conveyed in a clear and accessible manner. These assumptions serve as intermediate steps that enable the exploration of the prominent issues in modeling an adversary. The assumptions are as follows: 1) Available actions of the attacker and the probability distribution of attacker skill types are public knowledge. 2) Intents of the attacker are zero-sum. 3) The Attacker is greedy. 4) The worst case with respect to the attacker's computational ability is considered (i.e. it is only prohibitive for the defender to computer the belief function of the system).

5.6.1 Capabilities

An attacker's type will be defined as his skill level at disrupting a set of services. The skill level will be represented as a vector, where the jth component of the vector represents the attackers ability to disrupt service j. The values of the skill vector are in the range between 0 and 1, where 1 is expert skill and 0 is no skill at disrupting service s_j.

5.6.2 Intent

The ultimate objective of the adversary may be unknown. Although many cyber hackers aim to profit from their attacks, other hacker groups such as Anonymous employ denial of service attacks to make political statements and to seek publicity. Therefore, an approach that is used in this chapter to address the uncertainty of the adversaries objective is to consider the worst case with respect to the defender. In particular, the problem is modeled as a zero-sum game (i.e. a reward α for the attacker is a corresponding cost α to the defender). This zero-sum assumption allows performance guarantees to be made on security policies.

5.6.3 Patience

One can model an adversary as having a discount factor λ on its future payoff. A discount factor of $\lambda \approx 0$ would then indicate a greedy adversary that heavily discounts the future, while $\lambda \approx 1$ would be indicative of an adversary that is a long-term player that heavily discounts the present. An alternative interpretation of the discount factor is the patience of the adversary. Modeling a patient adversary introduces a probing complication that is absent in the greedy models considered in this chapter. Since a patient attacker may be willing to defer an immediate reward, he can consider choosing actions that may provide him with a better estimate of the current state of the security system. This improved estimate along with the adversaries knowledge of the HMM can then be used to make a better prediction of the future behavior of the system. The question that follows is when should he probe and when should he attack. This is an interesting question that will be considered in future research.

5.6.4 Beliefs & Strategies

An adversary's decision to disrupt a particular service s_j can be dependent on his ability to disrupt service s_j (i.e. his skill set), his payoff for disrupting service s_j, and his beliefs about the current operational state of each subsystem T_i. Computing the belief function of the current state of the system T is prohibitive. Consider the following example as an illustration. Suppose that a system T' is composed of N' subsystems that each have 10 operational states. The size of the state space of the system T' is the product of the individual subsystems and is equal to $10^{N'}$, and the beliefs are probabilities of state combinations of the subsystems, e.g. $\Delta(10^{N'})$. The worst case is assumed about the adversaries capabilities, which is that he can compute the belief function of the system, while the defender can only compute an estimate. Two techniques, quasi-beliefs and belief compression, will be introduced in the subsequent sections to address the computational challenges of the defender. These techniques can be used to calculate estimates of the belief function of the system T.

5.6.4.1 *Quasi-beliefs*

The main idea of quasi-beliefs (QB) is the following. Instead of computing the true beliefs of system T, e.g. $\Delta(10^N)$, an estimate of the beliefs of each subsystem T_i can be computed independently of the other subsystems T_{-i}. An issue that arises with computing independent beliefs of each subsystem is signal assignment. *Recall that the attacker only observes the collection of signals emitted from the subsystems. Therefore, he has uncertainty about the mapping between each signal and the subsystem that emitted the signal.* Algorithm 3, detailed below, provides a method for estimating the likely mapping between signals and subsystems.

Algorithm 3 Signal assignment

1: **procedure** SIGNALASSIGN
2: **initialize matrix** P_S
3: **while** size(P_S) > 0 **do**
4: find the maximum element of P_S
5: denote (i^*, m^*) as the position of the max element
6: assign signal y_{m^*} to subsystem T_{i^*}
7: remove row i^* and column m^* from matrix P_S
8: **end while**
9: **end procedure**

Note that for the matrix P_S at step 2 of Algorithm 3, each column corresponds to a signal y_m, each row corresponds to a subsystem T_i, and element (i, m) represents the probability that signal y_m was emitted from subsystem T_i. Also note that if there are more than one maximum element at step 4, the tie is broken by randomly selecting a maximum element. The signal assignment method is used in the quasi-belief greedy (QBG) algorithm (Algorithm 4) that is described below.

Algorithm 4 QBG Strategy

1: **procedure** QBGSTRATEGY
2: start with set of individual subsystem beliefs
3: **run** procedure SIGNALASSIGN
4: update individual beliefs using assignment
5: ***attack services with highest expected reward***
6: re-normalize beliefs given success/failure of attack
7: **end procedure**

5.6.4.2 Belief Compression

The point of departure is the statistical description of the HMM abstraction of the target system. Let $\mathbb{A} = \{a_1, \ldots, a_n\}$, $\mathbb{B} = \{b_1, \ldots, b_m\}$ denote the state and output space respectively. The statistics of the joint state and output process $\{(X_t, Y_t)\}_{t\in\mathbb{Z}_+}$ are encoded by the initial distribution vector π, where

$$\pi_i = \mathbf{Pr}[X_t = a_i],\ i \in \{1, \ldots, n\},$$

and a set of m transition matrices $\{M[y_1], \ldots, M[y_m]\}$ where

$$M[y_k]_{ij} = \mathbf{Pr}[Y_{t+1} = b_k, X_{t+1} = a_i \mid X_t = a_j],$$

for $k \in \{1, \ldots, m\}$ and $i, j \in \{1, \ldots, n\}$. Let $\mathbb{B}^*$ denote the set of all emitted finite strings of observation signals including the empty sting $\varnothing$. Let $v = v_k \ldots v_1$ stand for a string of length k. Let $1_n \in \mathbb{R}^n$ denote the vector whose entries are all 1. Introduce the function

$$p : \mathbb{B}^* \times \mathbb{R}_+^n \to [0, 1],$$

where

$$p[(v, \pi)] = 1_n^T\ M[v_k]\ \ldots\ M[v_1]\pi.$$

The function p is referred to as the probability function. It is used to compute the probability of observing a particular under initial distribution π, i.e.

$$p[(v, \pi)] = \sum_{i\in\{1,\ldots,n\}} \mathbf{Pr}[Y_k = v_k, \ldots, Y_1 = v_1 \mid X_o = a_i]\ \pi_i.$$

The value $p[(v, \pi)]$ is computed recursively using the rule

$$p[(v, \pi)] = 1_n^T \; H_k,$$

where $H_t = M[v_t] \; H_{t-1}$, $t \in \{1, \ldots, k\}$, and $H_0 = \pi$. Consider also the functions

$$p_{co} : \mathbb{Y} \times \mathbb{Y}^* \times \mathbb{R}_+^n \to [0, 1], \quad p_{cs} : \mathbb{X} \times \mathbb{Y}^* \times \mathbb{R}_+^n \to [0, 1],$$

referred to as the conditional output probability and conditional state probability function. The value $p_{co}[(b, v, \pi)]$ corresponds to the conditional probability of emitting the signal b given that the signal v has been observed under the initial distribution π, i.e.

$$p_{co}[(b, v, \pi)] = \sum_{i \in \{1, \ldots, n\}} \mathbf{Pr}[Y_{k+1} = b \mid Y_k = v_k, \ldots, Y_1 = v_1, X_0 = a_i] \; \pi_i.$$

Similarly the value $p_{cs}[(a, v, \pi)]$ corresponds to the conditional probability of being at state a given that the signal v has been observed under the initial distribution π, i.e.

$$p_{cs}[(a, v, \pi)] = \sum_{i \in \{1, \ldots, n\}} \mathbf{Pr}[X_k = a \mid Y_k = v_k, \ldots, Y_1 = v_1, X_0 = a_i] \; \pi_i.$$

The belief function is

$$\Pi : \mathbb{Y}^* \times \mathbb{R}_+^n \to \mathbb{R}_+^n,$$

where

$$\Pi[v, \pi]_i = p_{cs}[(a_i, v, \pi)], \quad i \in \{1, \ldots, n\}.$$

The value of the belief function is computed recursively by using the rule

$$\Pi[v, \pi] = \frac{H_k}{1_n^T \; H_k}.$$

At every time step the attacker chooses an action to maximize his instantaneous expected reward. For $s \in \mathbb{S}$ let $g[s] \in \mathbb{R}^n$ where $g[s]_i = r[s, a_i]$, $i \in \{1, \ldots, n\}$. Having observed the signal v and following a greedy strategy the attacker is faced with the optimization problem

$$\max_{s \in \mathbb{S}} < g[s], \Pi[v, \pi] > .$$

The notion of belief compression is associated with projecting the dynamics of the given HMM onto a lower dimensional manifold. Let $\hat{n} < n$, $V \in \mathbb{R}^{n \times \hat{n}}$, $U \in \mathbb{R}^{\hat{n} \times n}$ with $U\ V = I_{\hat{n}}$, so that $V\ U$ is a projection matrix. The parameters of a reduced complexity model are given by

$$\begin{aligned} \hat{c}^T &= 1_n^T\ V, \quad \hat{b} = U\ \pi, \\ \hat{A}[y] &= U\ M[y]\ V, \quad y \in \mathbb{Y}. \end{aligned}$$

Using the reduced complexity model one can determine a greedy strategy while performing the relevant calculations on a $\hat{n}$ dimensional space with obvious computational and storage advantages. In particular, consider the function

$$\hat{p} : \mathbb{Y}^* \times \mathbb{R}^n \to \mathbb{R},$$

where

$$\hat{p}[(v, \hat{b})] = \hat{c}^T\ \hat{A}[v_k]\ \ldots\ \hat{A}[v_1]\hat{b}.$$

The value $\hat{p}[(v, \hat{b})]$ is computed recursively using the rule

$$\hat{p}[(v, \hat{b})] = \hat{c}^T\ \hat{H}_k,$$

where $\hat{H}_t = \hat{A}[v_t]\ \hat{H}_{t-1}$, $t \in \{1, \ldots, k\}$, and $\hat{H}_0 = \hat{b}$. The function $\hat{p}$ is a low complexity surrogate for the probability function of the given HMM. Similarly consider the function

$$\hat{\Pi} : \mathbb{Y}^* \times \mathbb{R}^{\hat{n}} \to \mathbb{R}^{\hat{n}},$$

where

$$\hat{\Pi}[v, \hat{b}] = \frac{\hat{H}_k}{1_n^T\ \hat{H}_k}.$$

Let $\hat{g}[s] = V^T g[s]$, when employing the low complexity model the attacker is faced with the optimization problem

$$\max_{s \in \mathbb{S}} < \hat{g}[s], \hat{\Pi}[v, \pi] > .$$

The balanced truncation algorithm developed for HMM's in [54] will be employed to compute the compression matrix U and dilation matrix V. The reduction method is based on stable numerical linear algebra tools employing the singular value decomposition and is accompanied by an a priori bound to the approximation error. In other words, it leverages the favorable features of Hankel norm based reduction techniques for linear time invariant systems.

First one solves linear algebraic equations to obtain "gramian like" quantities $W_c, Wo \in \mathbb{R}^{n\times n}$ where $W_c, W_o \geq 0$,

$$W_o = \sum_{y\in\mathbb{Y}} M[y]^T\ W_o\ M[y] + 1_n^T\ 1_n,\ W_c = \sum_{y\in\mathbb{Y}} M[y]\ W_c\ M[y]^T + \pi\,\pi^T. \tag{75}$$

Denote by L_o, L_c the Cholesky factors of $W_o = L_o^T\ L_o$ and $W_c = L_c\ L_c^T$ and consider the SVD of $L_c^T\ L_o^T$,

$$L_c^T\ L_o^T = \left[\begin{array}{cc} \Psi^{(1)} & \Psi^{(2)} \end{array}\right] \left[\begin{array}{cc} \Sigma^{(1)} & 0 \\ 0 & \Sigma^{(2)} \end{array}\right] \left[\begin{array}{c} \Xi^{(1)^T} \\ \Xi^{(2)^T} \end{array}\right].$$

where $\Sigma^{(1)} = \mathrm{diag}[\sigma_1, \ldots, \sigma_{\hat{n}}]$, $\Sigma^{(2)} = \mathrm{diag}[\sigma_{\hat{n}+1}, \ldots, \sigma_n]$ and $\sigma_1 \geq \ldots \geq \sigma_{\hat{n}} > \sigma_{\hat{n}+1} \geq \ldots \geq \sigma_n > 0$. In the above notation

$$V = L_c[\psi_1^{(1)}\frac{1}{\sqrt{\sigma_1}}, \ldots, \psi_{\hat{n}}^{(1)}\frac{1}{\sqrt{\sigma_{\hat{n}}}}],\ U = \left[\begin{array}{c} \frac{1}{\sqrt{\sigma_1}}\xi_1^{(1)^T} \\ \vdots \\ \frac{1}{\sqrt{\sigma_{\hat{n}}}}\xi_{\hat{n}}^{(1)^T} \end{array}\right] L_o.$$

The following error bound controls the approximation of the given probability function:

$$\sqrt{\sum_{v\in\mathbb{Y}^*}(p[v,\pi] - \hat{p}[v,\hat{b}])^2} \leq 2(\sigma_{\hat{n}+1} + ... + \sigma_n).$$

The algorithm is demonstrated on a target system comprised from 3 subsystems used in the iCTF, with $|\mathbb{A}| = 1331$ and $|\mathbb{B}| = 680$. The singular values $\sigma_1, \ldots, \sigma_{1331}$ that control the error bound are illustrated in Figure 4. There is a clear cut-off behavior indicating that a choice of a reduced complexity model with 286 states is appropriate.

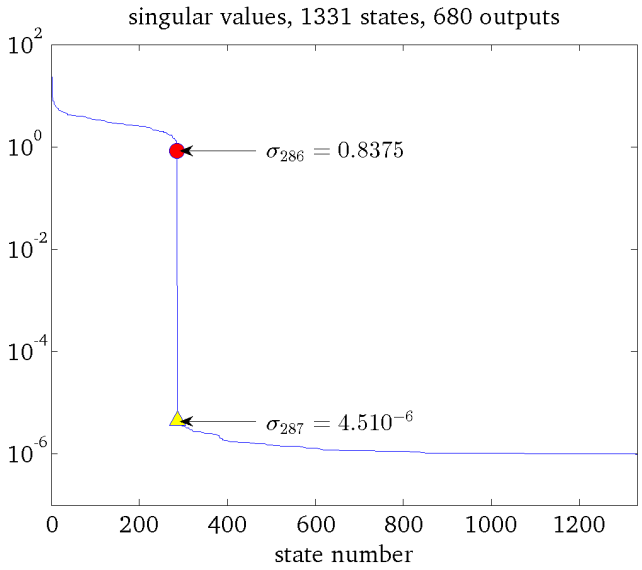


Figure 4: Singular values $\sigma_1, \ldots, \sigma_{1331}$ control the error bound.

Next it is demonstrated that the reduced complexity model delivers a very accurate approximation to the conditional output probability function. The vector

$$(p_{co}[b_1, v_k \ldots v_1, \pi], \ldots, p_{co}[b_{680}, v_k \ldots v_1, \pi])$$

conditioned on two trajectories $v_k \ldots v_1$ of length 1000 is depicted in Figure 5. The bottom row of Figure 5 corresponds to the exact model and the rows above it correspond to approximations computed using the reduced order model. In both cases a reduced order model of at most 286 states approximates this conditional probability within 0.1%.

One can use the reduced order model also to compute an approximation to the belief function of the system and subsequently solve the attacker's greedy optimization problem. A reduced order model of 499 states depicted in Figure 6 delivers a very accurate approximation to the belief function within 0.1% error. The belief function was computed for a trajectory $v_k \ldots v_1$ of length 1000. The bottom row corresponds to the exact model and the rows above it correspond to approximations computed using the reduced order model. The reduced order model led to the same choice of action in 96% of the instances when using greedy optimization.

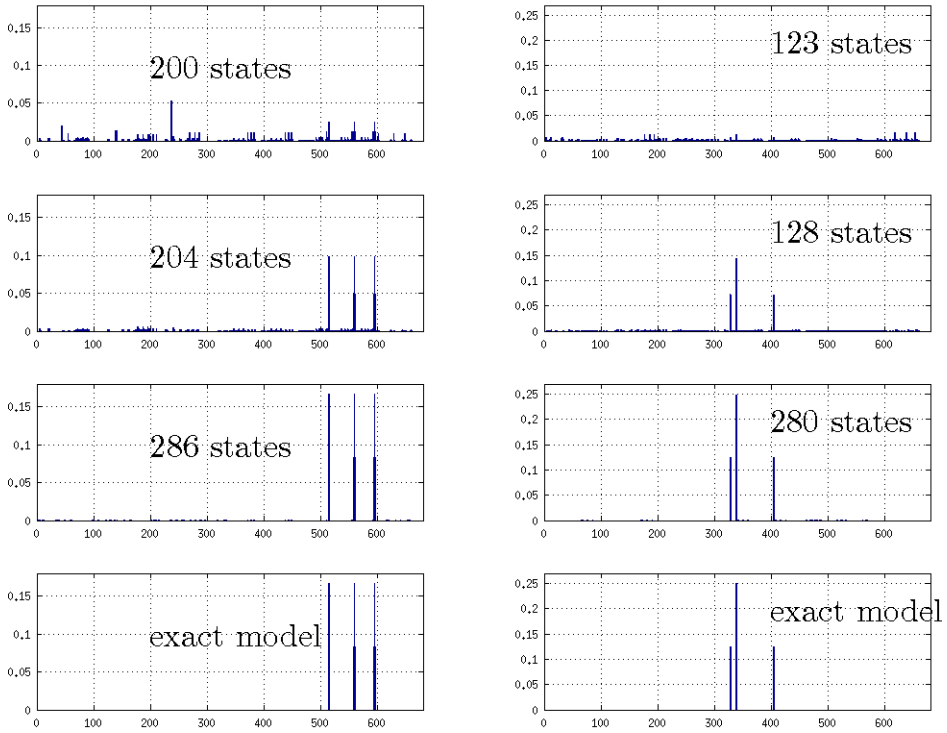


Figure 5: A comparison of the conditional probability of the approximate systems (first three rows) with respect to the exact system (bottom row).

5.7 CTF Security Game Formulation

In this section, iCTF will be formulated as a security game with asymmetric information. Recall that these games were discussed in Section 5.5. In the formulation that will be introduced in this section, it is assumed that only the attacker knows his opponent's type. Therefore, the information asymmetry of the iCTF security game is on the side of the attacker. The next section will proceed with a discussion of the game play. Strategies available to the players will then be discussed. Last, prominent issues that each must consider will be covered along with approaches to address those issues.

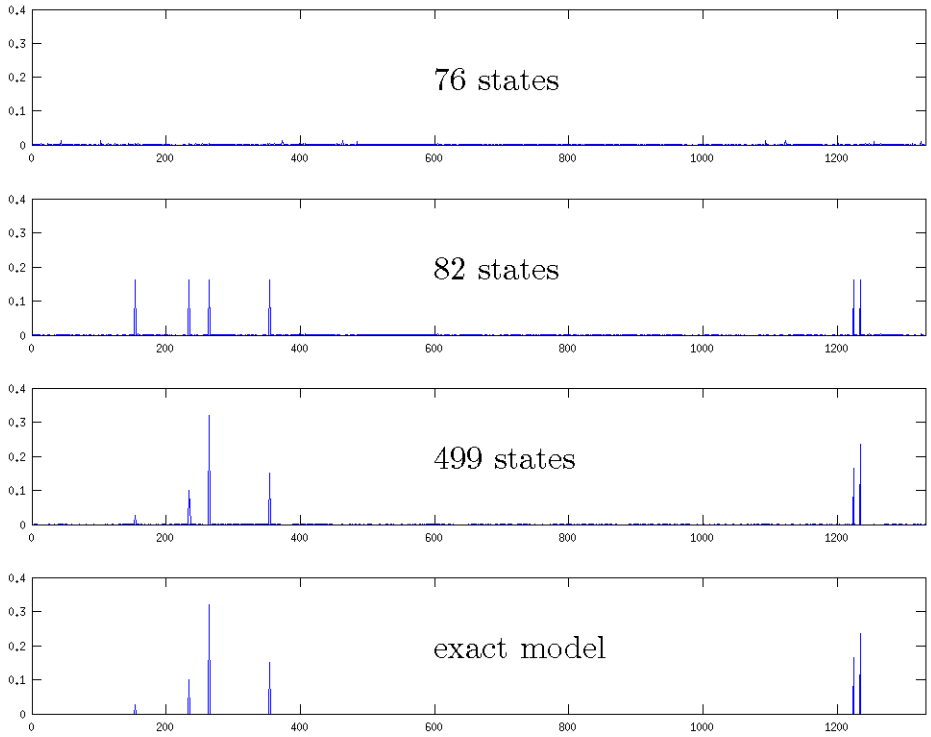


Figure 6: A reduced order model of 499 **states delivers a very accurate approximation to the belief function within** 0.1% **error.**

5.7.1 Game Play

5.7.1.1 One-shot game formulation

The one-shot game formulation of iCTF consists of two players, an attacker and the defender. An attacker of type k is selected by nature before the start of the game from a known distribution p of attacker types.[13] The attacker's objective in this game is to maximize his reward for attacking the security system T that was defined in Section 5.4.2.1.[14] The attacker maximizes his reward by disrupting critical services that are needed by the individual subsystems T_i of T. The attacker knows the specific services that are critical for each state of the subsystem. However, he has uncertainty about the current state of each subsystem T_i.

[13] Attackers differ in their ability to disrupt services, and an attacker's type is defined by its skill vector.
[14] Conversely, the defender's objective is to minimize its costs incurred by the attacker.

An attacker can improve his estimate of each subsystem's state by computing the belief function of the system T. Since computing the belief function is computationally prohibitive for systems with large state spaces, the defender can approximate the true belief function by considering a state estimation technique discussed in Section 5.6.4. It is assumed that the defender uses the quasi-beliefs in this chapter, although she could also use an alternative technique (i.e. belief compression). A particular greedy strategy that an attacker can consider is to disrupt services that maximize the expected payoff of an attacker of his type. This greedy strategy will be referred to as the honest strategy. Alternatively, an attacker can disrupt services that maximizes the payoffs of an attacker of another type. This strategy will be referred to as the dishonest strategy. It may initially seem irrational for an attacker to select a dishonest strategy because this strategy does not maximize his immediate payoff. However, it will be demonstrated in the next section that choosing a dishonest strategy can be rational and optimal.

The defender has finite resources that she can allocate to maintain the availability of critical services. Dedicating more resources to a particular service s_j decreases the likelihood that service s_j will be disrupted in the event that an attacker targets that service. At the start of iCTF, she can select how these resources are allocated, and the resource allocation decision remains fixed until the completion of iCTF. This resource allocation decision can be a function of the defender's belief about an attacker's type. In particular, she can chose to do a best response allocation with respect to an attacker's type or best respond to an attacker's type with some probability.

5.7.1.2 Repeated game formulation

It was assumed in the one-shot formulation that once a QBG policy is chosen by the attacker and a resource allocation technique is chosen by the defender at the beginning of the game, the selections by both players remain fixed throughout the duration of iCTF. In the repeated game formulation, however, both players can reevaluate their decisions at specific intervals. These intervals will be referred to as stages. At each stage, the security system S is reset

to its initial state. However, each player's knowledge about the other player's past actions is not reset. In fact, each player's knowledge changes over time, affects his/her beliefs, and introduces a dynamic aspect to this security game.

5.7.2 Player Concerns

In the previous section, it was discussed that each player's knowledge is dynamic and changes in time. An important objective of the attacker is to control the beliefs of the defender about his type because knowledge of the attacker's type can allow the defender to make decisions that cost the attacker. Controlling the defender's beliefs often involves deceptive play by the attacker. *An attacker of type k can elect to select a QBG strategy of an attacker of a different type. (i.e. $QBG_{k'}$ where $k \neq k'$).* Because of the potential for deception, it can be difficult for the defender to learn the attacker's true type. In the preceding sections, approaches for addressing each players concerns will be discussed.

5.7.2.1 Attacker

Recall that in Section 5.5.3.3 a formulation introduced by Aumann ad Maschler was discussed. This formulation can be used to model an attacker that optimally controls the beliefs of the defender and to determine the optimal game payoff. As part of the discussion, complexity issues that arose with using this formulation were mentioned. Policy-improvement strategies will be used to address the complexity issues by approximating the strategy selection of an optimal attacker. These policy-improvement strategies have error bounds on their performance with respect to optimal strategies, and the performance of the policy improvement methods converge asymptotically to optimal with respect to n, the number of stages in the game. The one-time policy-improvement method will first be discussed and then will be preceded by a discussion of the perpetual policy-improvement method.

In one-time policy improvement, an attacker strategizes for the *first stage* of the iCTF game while assuming that he will play in a non-revealing manner in all future stages.

Perpetual policy improvement is an extension of one-time policy improvement. In the perpetual policy-improvement method, the attacker strategizes for the current stage while assuming a non-revealing strategy in all future stages at *every stage*. The perpetual policy-improvement method involves solving an LP online, and the computational complexity of the LP is constant with respect tho the number of stages of the game. Below is the perpetual policy improvement algorithm.

Algorithm 5 Perpetual policy improvement

1: **procedure** PERPPOLICYIMPROVE
2: **initialize:** set $p_1 = p$
3: **for** $m = 1 \rightarrow N$ **do**
4: compute $\hat{x}_m$ by solving one-time policy improvement LP with p_m
5: select a move s for attacker type k using mixed strategy $\hat{x}_m^k$
6: update beliefs vector (i.e.$p_{m+1} = p^+(p, \hat{x}, s)$)
7: **end for**
8: **end procedure**

5.7.2.2 Defender

Learning an attacker's true type can be challenging because he can play in a deceptive manner [36]. If the defender knew the mixed strategy x^k for each type of attacker k, then learning an attacker's type would be straightforward because the defender could follow a standard Bayesian-update approach. Unfortunately, in the actual play of the game, the defender does not know each attacker's mixed strategy because this information is private. Another approach that the defender can consider is solving a linear program to compute an optimal defensive strategy.[15] However, the complexity of the LP is exponential with respect to n, the number of stages of the game. A payoff-based heuristic for learning an attacker's type will be introduced that is computationally tractable for arbitrary n and only depends

[15] Ponssard and Sorin [42] showed that zero-sum repeated games of incomplete information can be formulated as linear programing problems to compute optimal strategies.

on the information that is available to the defender. This information is namely the history of the attacker's actions.

The main idea behind the payoff-based heuristic is as follows. The defender's belief of an attacker of type k will be correlated with the actual game payoff of the attacker. After each stage, the defender keeps track of what the overall game payoff would be for each type of attacker. The game payoff for an attacker of type k at stage n, given history h_n will be denoted $\tilde{\gamma}_n^k(h_n)$. This payoff $\tilde{\gamma}_n^k(h_n)$ will be compared to the best possible payoff that a type k attacker can achieve. The best possible payoff will be denoted by $|M^k|$, where $|M^k| := \max_{i,j} M_{i,j}^k$. *Without loss of generality, it will be assumed that each matrix M^k is scaled with values ranging from 0 to 1.* Let

$$\xi^k(h_n) = \frac{\tilde{\gamma}_n^k(h_n)}{|M^k|} \tag{76}$$

be a measure of the likelihood that an attacker is of type k given history h_n. The belief update procedure is then

$$\tilde{p}_{n+1}^k(h_n) = \tilde{p}_n^k \frac{\xi^k(h_n)}{\bar{\xi}(h_n)}, \tag{77}$$

where $\bar{\xi}(h_n) = \sum_{k \in K} \tilde{p}_n^k \xi^k(h_n)$. To compute a best response strategy $\tilde{y}^*$ for the defender given the approximate belief $\tilde{p}_n$ at stage m, solve the optimization equation

$$\tilde{y}_m^* = \arg\min_{y_m} \max_{x_m} \sum_{k \in K} \tilde{p}_m^k x_m^k M^k y_m. \tag{78}$$

5.8 Simulation

5.8.1 Game Setup

As usual, this game consists of two players, an attacker and a defender. In this example, it is assumed that there are two types of attackers (i.e. type I and type II) and each attacker is uniquely defined by his skill vector. The probability distribution of attacker types is uniform (i.e. $p^k = \frac{1}{2}$ for $k = 1, 2$), and there are two stages in this game. Matrix payoffs for the attacker types are

	BR_1	BR_2
QBG_1	23	375
QBG_2	−92	69

Type I

	BR_1	BR_2
QBG_1	−6	−28
QBG_2	128	−20

Type II

(79)

Note that an attacker of type I has the option of playing as his type by selecting QBG_I or playing deceptively by selecting QBG_{II}. Similarly, a type II attacker can opt to play either honestly or deceptively. The defenders available actions are to play a best-response resource-allocation strategy for a specific attacker type (i.e BR_I or BR_{II}).

5.8.2 Discussion

The performance of four attacker strategies will be discussed in this section. These strategies are dominant strategy, non-revealing strategy, one-time policy improvement, and perpetual policy improvement. In the one-shot game, the optimal strategy for an attacker is to behave as his type by selecting his dominant strategy. However, for games where $n > 1$, this can be a suboptimal strategy because it can reveal the attackers true type to the defender and cost the attacker the informational advantage. Specifically, in the two stage game, the attacker can achieve a better payoff by selecting the perpetual policy-improvement strategy. For games where N is large, the dominant strategy has the worst performance out of the four strategies and the policy improvement strategies have the best performance.

5.8.2.1 Two-stage game

An optimal non-revealing strategy requires the attacker, regardless of his type, to play as a type I attacker with probability .70 and to play as a type II attacker with probability .30 at each stage. This strategy rewards the attacker with a payoff of 18 and has the worst performance of the four strategies in the two-stage game.[16] One-time policy improvement

[16]There are games where playing non-revealing is optimal for all n.

performs better by guaranteeing an expected payoff of 27. The two strategies differ conceptually only at the first stage, where the one-time policy-improvement strategy is dependent on the attacker's type. A type I attacker plays deceptively at stage $m = 1$ with probability .92, while a type II attacker plays honestly with probability 1. At $m = 2$, an attacker plays as a type II attacker with probability .96, which is independent of his type. An attacker that chooses to use his dominant strategy, which requires him to play honestly, at each stage of the game yields the attacker an expected payoff of 39, which outperforms the two previously mentioned strategies. Perpetual policy improvement yields the attacker the highest reward, 53, of the four strategies in consideration. At the first stage of perpetual policy improvement, the attacker plays the same way he would have played had he chosen one-time policy improvement. However, the key difference is at the second stage. Instead of playing non-revealing as with the former strategy, the attacker behaves as his type in the second stage of perpetual policy improvement.

5.8.2.2 N-stage game

The performance of the four strategies in the two-stage case was discussed in the previous section. The performance of these strategies as N grows large will now be examined. The expected payoff for the dominant strategy converges asymptotically to 2 and has the worst performance of the four strategies for large N. This asymptotic converge happens because the defender can readily learn the attacker's type, since the attacker does not play deceptively. The defender can then use this knowledge to select a resource allocation scheme that is a best response to his type. An optimal non-revealing strategy performs better than the dominant strategy because the defender is unable to learn any additional information about the attacker after observing his action at each stage. As a consequence, the defender has uncertainty about which resource allocation scheme will perform best against the attacker. An attacker who chooses this strategy can therefore guarantee a payoff of 18 at every stage. An immediate consequence of this guarantee is that an attacker can achieve a game payoff of 18 for games of any length. Policy improvement methods have the best performance of

the four strategies for large N. Both methods have identical behavior and converge asymptotically to optimal and yields a payoff of 23. At stage one of the policy improvement methods, the attacker behaves deceptively with some probability that is type dependent. For all stages thereafter, the attacker plays a non-revealing strategy that is independent of his type.

5.9 Remarks (Stochastic Game Extensions)

So far we have considered the cyber scenario where an attacker is randomly selected by nature. The attacker selection along with the corresponding skill set of the attacker remains fixed over all stages. We will now consider the scenario where the attacker's skill set can change during game play. Specifically, we will formulate the iCTF problem as a stochastic game and use the results of Chapter 4 to analyze the problem.[17]

5.9.1 Stochastic game formulation

The scenario we will consider is the following. In state ω_1, each attacker will have the identical skill sets that was presented in the repeated game. The key difference between that example and this example is that the skill set of the type I attacker will improve if he selects the action QBG_1; his skill at attacking service S_0 will improve from .15 to 1. This skill change will be represented by the game transitioning from state ω_1 to ω_2. The Payoff structure for the stochastic cyber-security game is the following:

[17]Explicit knowledge of the stochastic game results are not necessary for this section. We will use the results to compute the performance, but we will not go into any detail about the specific computation.

$k = 1$

	BR_1	BR_2	ω^+
QBG_1	23	375	ω_2
QBG_2	−92	69	–

$k = 2$

	BR_1	BR_2	ω^+	
QBG_1	−6	−28	ω_2	ω_1
QBG_2	128	−20	–	

	BR_1	BR_2	ω^+
QBG_1	275	607	–
QBG_2	−23	121	–

	BR_1	BR_2	ω^+	
QBG_1	−15	−17	–	ω_2
QBG_2	167	−15	–	

5.9.2 Discussion

We will consider a setup with the following paramters. The probability that an attacker of a particular type is selected will be assumed to be uniform. The game will be over two stages. In the game with these parameters, an attacker can guranatee an expected payoff of 9 if he plays an action (QBG_2) that keeps him in state ω_1. Alternatively, the attacker can change the state to ω_2 by choosing the action QBG_1. He can gurantee a payoff of at least 87. Note that the specific policy to achieve this payoff is perpetual policy improvement. For games where the number of stages N is large, the payoff for policy improvement converges to 173.

CHAPTER 6

CONCLUSION

The overarching theme of this thesis is to address computational complexity issues in the domains of Game Theory and cyber security. In Game Theory, computing optimal policies for asymmetric games are prohibitive. In cyber security, computational challenges have inhibited the realization of reliable cyber-attack forecasting. We address these issues by considering *computable* suboptimal polices with provable performance guarantees that are based on the concepts of model predictive control.

6.1 Asymmetric Information Games

Game theory has seen wide-spread adoption as a tool to model conflict-resolution scenarios and predict the likely strategies of decision makers. A classic conflict-resolution scenario is the Prisoner's dilemma, which captures how rational decision makers can resolve a conflict by choosing actions that serve there selfish interest. It is assumed in the classic version of Prisoner's dilemma that all information is public, but what happens when one of the decision makers has private information about the underlying state of the world? Works that includes [10], [24] have been highly influential in formulating this scenario as a strategic game and characterizing the equilibrium. Unfortunately, computing the equilibrium strategy for these games is computational prohibitive because of the exponential growth in complexity as the number of game stages increases.

The computational complexity problems that this class of games presents has not been adequately addressed in the literature. Work that includes [36], [5], [6], [37] makes some headway but is largely limited to exploring special cases. We address the complexity issues of stochastic and repeated games by revisiting the dynamic programing formulation, introduced by Aumann and Maschler, to pinpoint the cause of the complexity and consider

ways of reducing it. A key observation is that the cost-to-go function of the DP is the primary contributor to the exponential computational growth. We reduce the complexity of the cost-to-go by assuming non-revealing policies going forward instead of assuming optimal policies. We then apply model predictive control concepts to the problem that enable us to solve the problem as a linear program online at each stage of the game, where the complexity of the LP remains constant for arbitrary game stages. We are then able to prove that the suboptimal policies have bounds on their performance.

6.2 Cyber Security

Much research has been dedicated to formulating cyber-security problems as strategic games to forecast a cyber-attack and predict the likely behavior of key decision makers. This research includes the work of [55], [56], [57], [58], [59], [60], [61]. Impediments to the realization of cyber-attack forecasting includes modeling and computational challenges. Specifically, it is unclear of what the appropriate method for formulating a cyber security problem as a game should be [52]. It can also be computationally difficult to analyze a strategic game because of the complexity challenges of computing equilibrium strategies.

6.2.1 Modeling Cyber-security Problems

Our work in this thesis largely involves addressing computational challenges. However, in Chapter 5, we consider a capture-the-flag challenge problem developed by security researches at UCSB [52] and formulate this cyber-security scenario in a game theoretic framework. The approaches that we used to frame the problem as a strategic game could also be used for other problems. Since how something is modeled can significantly be influential in how difficult it is to analyze the model, choosing an appropriate model is also very important with respect to computational considerations.

6.2.2 Complexity Reduction

As was pointed out previously, there can be an intimate relationship between a particular model that is chosen to model a security system and the computation required to analyze the model. Therefore, how much one can reduce the overall computational complexity is a function of the model. Given a game-theoretic formulation of the model, however, we introduce methods in this thesis that can be applied to the strategic game formulation that yields suboptimal policies that can approximate the behavior of an optimal attacker. Having the ability to to predict an attacker's behavior can better assist security researchers in discovering vulnerabilities in security systems. This ability can also enable security professionals and practitioners to be more proactive instead of reactive with respect to cyber security.

6.3 Future Work

In this dissertation, we address the complexity challenges of forecasting a cyber-attack by introducing complexity reduction techniques. The security system model that we used for the cyber-attack forecasting was an experimental model constructed by security researchers. We would like to next consider a real-world security system and model the system in a game-theoretic framework. Although taking this next step in modeling a real-world system is also challenging, this next step can potentially provide rich insights into the performance of the system and unearth system vulnerabilities that may have been previously unknown and perhaps not considered. Modeling a real-world system could also provide security researchers with tangible proof that formulating cyber-security problems in an analytical framework has significant value to security practitioners. Incorporating the reduction techniques presented in this research on a real-world system can make significant inroads in demonstrating that cyber-attack forecasting can be a practical proactive security approach.

REFERENCES

[1] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, pp. 120–126, 1978.

[2] M. Hasan, N. Prajapati, and S. Vohara, "Case study on social engineering techniques for persuasion," 2010.

[3] M. A. Bishop, *The Art and Science of Computer Security*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002.

[4] J.-P. Ponssard and S. Sorin, "Optimal behavioral strategies in zero-sum games with almost perfect information," *Mathematics of Operations Research*, vol. 7, no. 1, pp. pp. 14–31, 1982.

[5] V. C. Domansky and V. L. Kreps, "Eventually revealing repeated games of incomplete information," *International Journal of Game Theory*, vol. 23, pp. 89–109, 1994.

[6] S. Zamir, "Repeated games of incomplete information: Zero-sum," *Handbook of Game Theory*, vol. 1, pp. 109–154, 1999.

[7] S. Sorin, *A First Course on Zero-Sum Repeated Games*. Springer, 2002.

[8] R. Aumann, *Mixed and behavior strategies in infinite extensive games*. Princeton University, 1961.

[9] D. Rosenberg, E. Solan, and N. Vieille, "Stochastic games with a single controller and incomplete information," tech. rep., Northwestern University, May 2002.

[10] R. J. Aumann and M. Maschler, *Repeated Games with Incomplete Information*. MIT Press, 1995.

[11] N. Krishnamurthy, T. Parthasarathy, and G. Ravindran, "Communication complexity of stochastic games," in *Game Theory for Networks, 2009. GameNets '09. International Conference on*, pp. 411–417, 2009.

[12] R. Colbaugh and K. Glass, "Proactive defense for evolving cyber threats," in *Intelligence and Security Informatics (ISI), 2011 IEEE International Conference on*, pp. 125–130, 2011.

[13] M. Jones, G. Kotsalis, and J. Shamma, "Cyber-attack forecast modeling and complexity reduction using a game-theoretic framework," in *Control of Cyber-Physical Systems* (D. C. Tarraf, ed.), vol. 449 of *Lecture Notes in Control and Information Sciences*, pp. 65–84, Springer International Publishing, 2013.

[14] M. E. OConnell, "Cyber security without cyber war," *Journal of Conflict and Security Law*, vol. 17, no. 2, pp. 187–209, 2012.

[15] G. J. Mailath and L. Samuelson, *Repeated Games and Reputations: Long-Run Relationships*. Oxford University Press, 2006.

[16] M. J. Osborne, *An Introduction to Game Theory*. Oxford University Press, USA, Aug. 2003.

[17] D. Fudenberg and J. Tirole, *Game Theory*. MIT Press, Aug. 1991.

[18] J. Nash, *Non-Cooperative Games*. PhD thesis, Priceton University, 1950.

[19] J. Von Neumann and O. Morgenstern, *The theory of games and economic behavior*. Princeton, 3 ed., 1990.

[20] T. H. Kjeldsen, "John von neumann's conception of the minimax theorem: A journey through different mathematical contexts," *Archive for History of Exact Sciences*, vol. 56, no. 1, pp. 39–68, 2001.

[21] J.-P. Ponssard, "A note on the l-p formulation of zero-sum sequential games with incomplete information," *International Journal of Game Theory*, vol. 4, no. 1, pp. 1–5, 1975.

[22] D. Fudenberg and E. Maskin, "The Folk Theorem in Repeated Games with Discounting or with Incomplete Information," *Econometrica*, vol. 54, pp. 533–554, May 1986.

[23] W. Poundstone, *Prisoner's Dilemma*. New York, NY, USA: Doubleday, 1st ed., 1993.

[24] J. C. Harsanyi, "Games with incomplete information played by "bayesian" players, i-iii. part iii. the basic probability distribution of the game," *Management Science*, vol. 14, no. 7, pp. pp. 486–502, 1968.

[25] J.-F. Mertens and S. Zamir, "Formulation of bayesian analysis for games with incomplete information," *International Journal of Game Theory*, vol. 14, no. 1, pp. 1–29, 1985.

[26] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 2nd ed., 2000.

[27] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY, USA: John Wiley & Sons, Inc., 1st ed., 1994.

[28] R. Bellman, *Dynamic Programming*. Princeton, NJ, USA: Princeton University Press, 1 ed., 1957.

[29] R. E. Bellman and S. E. Dreyfus, *Applied Dynamic Programming*. Princetown University Press, 1962.

[30] D. P. Bertsekas and S. E. Shreve, *Stochastic Optimal Control: The Discrete-Time Case*. Athena Scientific, 2007.

[31] D. P. Bertsekas, "Dynamic programming and suboptimal control: A survey from adp to mpc," in *CDC Proceedings*, 2005.

[32] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Athena Scientific, 1st ed., 1996.

[33] D. P. Bertsekas and D. A. Castaon, "Rollout algorithms for stochastic scheduling problems," *Journal of Heuristics*, pp. 89–108, 1999.

[34] J. Rawlings, "Tutorial overview of model predictive control," *Control Systems, IEEE*, vol. 20, no. 3, pp. 38–52, 2000.

[35] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, pp. 789–814, June 2000.

[36] M. Heur, "Optimal strategies for the uninformed player," *International Journal of Game Theory*, vol. 20, pp. 33–51.

[37] A. Gilpin and T. Sandholm, "Solving two-person zero-sum repeated games of incomplete information," *International Joint Conference on Autonomous Agents and Multiagent Systems*, vol. 2, pp. 903–910, 2008.

[38] S. Zamir, "On the relation between finitely and infinitely repeated games with incomplete information," *International Journal of Game Theory*, vol. 23, pp. 179–198.

[39] A. g. J Cruz, M. Simaan and Y. Liu, "Moving horizon nash strategies for a military operation," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 348, no. 3, 2002.

[40] W. v. d. Broek, "Moving horizon control in dynamic games," Discussion Paper 1999-07, Tilburg University, Center for Economic Research, 1999.

[41] H. S. Chang and S. I. Marcus, "Two-person zero-sum markov games: receding horizon approach.," *IEEE Trans. Automat. Contr.*, vol. 48, no. 11, pp. 1951–1961, 2003.

[42] J. Ponssard and S. Sorin, "The l-p formulation of finite zero-sum games with incomplete information," *International Journal of Game Theory*, vol. 9, pp. 99–105, 1999.

[43] T. Raghavan and Z. Syed, "A policy-improvement type algorithm for solving zero-sum two-person stochastic games of perfect information," *Mathematical Programming*, vol. 95, no. 3, pp. 513–532, 2003.

[44] T. Raghavan, "Finite-step algorithms for single-controller and perfect information stochastic games," in *Stochastic Games and Applications* (A. Neyman and S. Sorin, eds.), vol. 570 of *NATO Science Series*, pp. 227–251, Springer Netherlands, 2003.

[45] S. Lakshmivarahan and K. S. Narendra, "Learning algorithms for two-person zero-sum stochastic games with incomplete information," *Mathematics of Operations Research*, vol. 6, no. 3, pp. pp. 379–386, 1981.

[46] P. S. Sastry, V. V. Phansalkar, and M. Thathachar, "Decentralized learning of nash equilibria in multi-person stochastic games with incomplete information," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 24, no. 5, pp. 769–777, 1994.

[47] L. S. Shapley, "Stochastic Games," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 39, no. 10, pp. 1095–1100, 1953.

[48] E. Kohlberg, "Repeated games with absorbing states," *The Annals of Statistics*, vol. 2, no. 4, pp. pp. 724–738, 1974.

[49] J. F. Mertens and A. Neyman, "Stochastic games," *International Journal of Game Theory*, 1981.

[50] T. Alpcan and T. Başar, "An intrusion detection game with limited observations," in *12th Int. Symp. on Dynamic Games and Applications*, (Sophia Antipolis, France), July 2006.

[51] X. Z. You and Z. Shiyong, "A kind of network security behavior model based on game theory," in *Parallel and Distributed Computing, Applications and Technologies, 2003. PDCAT'2003. Proceedings of the Fourth International Conference on*, pp. 950–954, 2003.

[52] A. Doupé, M. Egele, B. Caillat, G. Stringhini, G. Yakin, A. Zand, L. Cavedon, and G. Vigna, "Hit 'em where it hurts: a live security exercise on cyber situational awareness," in *Proceedings of the 27th Annual Computer Security Applications Conference*, ACSAC '11, (New York, NY, USA), pp. 51–61, ACM, 2011.

[53] M. Jones and J. Shamma, "Policy improvement for repeated zero-sum games with asymmetric information," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pp. 7752–7757, 2012.

[54] G. Kotsalis, A. Megretski, and M. Dahleh, "Balanced truncation for a class of stochastic jump linear systems and model reduction for hidden markov models," *Automatic Control, IEEE Transactions on*, vol. 53, no. 11, pp. 2543–2557, 2008.

[55] K. Vamvoudakis, J. Hespanha, B. Sinopoli, and Y. Mo, "Adversarial detection as a zero-sum game," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pp. 7133–7138, 2012.

[56] R. A. Miura-Ko, B. Yolken, J. Mitchell, and N. Bambos, "Security decision-making among interdependent organizations," in *Proceedings of the 2008 21st IEEE Computer Security Foundations Symposium*, CSF '08, (Washington, DC, USA), pp. 66–80, IEEE Computer Society, 2008.

[57] L. A. Gordon, M. P. Loeb, and W. Lucyshyn, "Sharing information on computer systems security: An economic analysis," *Journal of Accounting and Public Policy*, vol. 22, no. 6, pp. 461–485, 2003.

[58] L. Jiang, V. Anantharam, and J. Walrand, "How bad are selfish investments in network security?," *Networking, IEEE/ACM Transactions on*, vol. 19, no. 2, pp. 549–560, 2011.

[59] E. Gal-Or and A. Ghose, "The economic incentives for sharing security information," *Information Systems Research*, vol. 16, no. 2, pp. 186–208, 2005.

[60] Y. Liu, C. Comaniciu, and H. Man, "A bayesian game approach for intrusion detection in wireless ad hoc networks," in *Proceeding from the 2006 workshop on Game theory for communications and networks*, GameNets '06, (New York, NY, USA), ACM, 2006.

[61] S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya, and Q. Wu, "A survey of game theory as applied to network security," in *Proceedings of the 2010 43rd Hawaii International Conference on System Sciences*, HICSS '10, (Washington, DC, USA), pp. 1–10, IEEE Computer Society, 2010.

[62] D. Blackwell, "An analog of the minimax theorem for vector payoffs.," *Pacific Journal of Mathematics*, vol. 1956, no. 1, pp. 1–8, 1956.

[63] Y. Freund and R. E. Schapire, "Game theory, on-line prediction and boosting," in *Proceedings of the ninth annual conference on Computational learning theory*, COLT '96, (New York, NY, USA), pp. 325–332, ACM, 1996.

[64] J.-F. Mertens and S. Zamir, "The value of two-person zero-sum repeated games with lack of information on both sides," in *Institute of Mathematics, The Hebrew University of Jerusalem*, pp. 405–433, 1970.

[65] J.-F. MERTENS, "The speed of convergence in repeated games with incomplete information on one side," CORE Discussion Papers 1995006, Universit catholique de Louvain, Center for Operations Research and Econometrics (CORE), Jan. 1995.

[66] D. Mayne and H. Michalska, "Receding horizon control of nonlinear systems," *Automatic Control, IEEE Transactions on*, vol. 35, pp. 814 –824, jul 1990.

[67] W. H. Kwon and S. Han, "Receding horizon schemes for controls, estimations, and optimizations," in *Control, Automation and Systems, 2007. ICCAS '07. International Conference on*, pp. xlv –lv, oct. 2007.

[68] H. Chen, C. Scherer, and F. Allgower, "A game theoretic approach to nonlinear robust receding horizon control of constrained systems," in *American Control Conference, 1997. Proceedings of the 1997*, vol. 5, pp. 3073 –3077 vol.5, jun 1997.

[69] B. Abramson, "Expected-outcome: a general model of static evaluation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, no. 2, pp. 182–193, 1990.

[70] G. Tesauro and G. R. Galperin, "On-line policy improvement using monte-carlo search," in *NIPS'96*, pp. 1068–1074, 1996.

[71] T. Sun, Q. Zhao, and P. Luh, "A rollout algorithm for multichain markov decision processes with average cost," in *Positive Systems* (R. Bru and S. Romero-Viv, eds.), vol. 389 of *Lecture Notes in Control and Information Sciences*, pp. 151–162, Springer Berlin Heidelberg, 2009.

[72] M. Morari and J. H. Lee, "Model predictive control: Past, present and future," *Computers and Chemical Engineering*, vol. 23, pp. 667–682, 1999.

[73] R. Findeisen, L. Imsland, F. Allgower, and B. A. Foss, "State and output feedback nonlinear model predictive control: An overview," *European Journal of Control*, vol. 9, pp. 190–205, 2003.

[74] M. J.M., *Predictive Control with Constraints*. Prentice-Hall, 2002.

[75] N. Nisan, M. Schapira, G. Valiant, and A. Zohar, "Best-response mechanisms.," in *ICS* (B. Chazelle, ed.), pp. 155–165, Tsinghua University Press, 2011.

[76] E. Kohlberg, "Optimal strategies in repeated games with incomplete information," *International Journal of Game Theory*, vol. 4, no. 1, pp. 7–24, 1975.

[77] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou, "The complexity of computing a nash equilibrium," in *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, STOC '06, (New York, NY, USA), pp. 71–78, ACM, 2006.

[78] C. Kruegel and G. Vigna, "You are what you include: Large-scale evaluation of remote javascript inclusions," in *Conference on Computer and Communications Security (CCS)*, (Raleigh, NC, USA), ACM, October 2012.

[79] A. Doupe, L. Cavedon, C. Kruegel, and G. Vigna, "Enemy of the State: A State-Aware Black-Box Vulnerability Scanner," in *Proceedings of the USENIX Security Symposium (USENIX)*, (Bellevue, WA), August 2012.

[80] B. Stone-Gross, M. Cova, B. Gilbert, L. Cavallaro, M. Szydlowski, C. Kruegel, G. Vigna, and R. Kemmerer, "Your Botnet is My Botnet: Analysis of a Botnet Takeover," in *Proceedings of the Computer and Communications Security Conference (CCS)*, (Chicago, IL), November 2009.

[81] Q. Zhu and T. Basar, "Dynamic policy-based ids configuration," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pp. 8600–8605, 2009.

[82] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*. New York, NY, USA: Cambridge University Press, 2007.

[83] J. Pita, M. Jain, F. Ordez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus, "Using game theory for los angeles airport security.," *AI Magazine*, vol. 30, no. 1, pp. 43–57, 2009.

[84] E. Guillen, D. Padilla, and Y. Colorado, "Weaknesses and strengths analysis over network-based intrusion detection and prevention systems," in *Communications, 2009. LATINCOM '09. IEEE Latin-American Conference on*, pp. 1–5, 2009.