

The Human Element: Addressing Human Adversaries in Security Domains

by

James Pita

A Dissertation Presented to the
FACULTY OF THE USC GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(COMPUTER SCIENCE)

December 2012

Copyright 2012

James Pita

Acknowledgments

An individual and special thank you belongs to my adviser Milind Tambe. I cannot begin to thank you enough for your steadfast effort, determination, and dedication to each of your students. Not only are you the exemplar of what an adviser should be, but you are genuinely cherished by anyone who has had the pleasure to work with you. You have made this experience altogether remarkable due to your outstanding guidance and more importantly your sincere friendship. I would also like to thank my committee members for helping to guide my research and think beyond it: Jonathan Gratch, Richard John, Sarit Kraus, Stacy Marsella, and Nicholas Weller. I would particularly like to thank Sarit Kraus for her unparalleled insights, guidance, and assistance throughout my career and Richard John for helping me to expand my understanding of experimental approaches. Furthermore, I would like to thank my co-authors over the years: Bo An, Harish Bellamane, Shane Cullen, Manish Jain, Richard John, Christopher Kiekintveld, Sarit Kraus, Jun-young Kwak, Reuma Magori-Cohen, Rajiv Maheswaran, Janusz Marecki, Thanh Nguyen, Fernando Ordóñez, Praveen Paruchuri, Christopher Portway, Shyamsunder Rathi, Michael Scott, Eric Shieh, Erin Steigerwald, Milind Tambe, Jason Tsai, Craig Western, Rong Yang, and Zhengyu Yin. Your dedicated efforts, assistance, guidance, and hard work made this experience exceptionally better.

Beyond my mentors and collaborators, I would like to thank CREATE, the Los Angeles World Airport (LAWA) police, and the Transportation Security Administration for giving me the opportunity to work on real-world problems that have a direct impact on the community. A special thank you to Erroll Southers for tirelessly promoting ARMOR as a viable approach for critical security problems, Ernest Cruz for his countless hours developing the original ARMOR software package, and Shane Cullen and Erin Steigerwald for their efforts in developing the GUARDS system. I also want to thank my colleagues at USC and the greater TEAMCORE community. You have all helped make these past years a special experience for me and I will never forget all the times we have shared and all the help and guidance you have given me. A special thanks goes to Janusz Marecki for all the laughs and advice over the years. I would also like to thank God for this tremendous opportunity. Finally, more than thank you goes to my mother Diane Pita, father Eugene Pita, and brother Michael Pita. Thank you for a lifetime of support in everything and anything that I do. Without your unconditional love and support I would not have been able to get where I am today. Thank you for making me push my limits, explore the world around me, and expand my horizons. Thank you for all the sacrifices you have made to get me here and for your never ending guidance, effort, support, and love. Thank you for being the cornerstone of my life.

Table of Contents

Acknowledgments	ii
List of Figures	vii
Abstract	ix
Chapter 1: Introduction	1
1.1 Problem Addressed	1
1.2 Contributions	4
1.2.1 COBRA/MATCH	5
1.2.2 Security Circumvention Games	8
1.3 Guide to Thesis	10
Chapter 2: Background	11
2.1 Stakelberg Games	11
2.2 Bayesian Stackelberg Games	13
2.3 Strong Stackelberg Equilibrium	15
2.4 DOBSS and Baseline Algorithms	16
2.4.1 DOBSS	17
2.4.2 UNIFORM	20
2.4.3 MAXIMIN	21
2.5 BRQR	21
2.6 Security Stackelberg Games	23
2.7 Los Angeles International Airport	26
2.8 Human Subjects	28
Chapter 3: Related Work	32
3.1 Computing Optimal Stackelberg Equilibria	32
3.1.1 Efficient Solutions to general Bayesian Stackelberg games	32
3.1.2 Efficient Solutions for Large-Scale Security Games	35
3.2 Computing Robust Strategies	37
3.3 Addressing Suboptimal Decisions	38

Chapter 4: COBRA Algorithm	42
4.1 Key Ideas	44
4.1.1 Bounded Rationality	44
4.1.2 Anchoring Theory	45
4.2 Robust Algorithm	48
4.2.1 COBRA(0, ϵ)	48
4.2.2 COBRA(α , 0)	50
4.2.3 COBRA(α , ϵ)	51
4.2.4 Complexity	52
4.3 Equivalences Between Models	53
4.4 Experiment Purpose, Design, and Results	56
4.4.1 Purpose of this Study	56
4.4.2 Experimental Design	57
4.4.2.1 Participants	58
4.4.2.2 Reward Structure	58
4.4.2.3 Observability Conditions	60
4.4.2.4 Algorithms and Parameters	62
4.4.2.5 Experimental Procedure	66
4.4.3 Experimental Results	68
4.4.3.1 Key Observations	70
4.4.3.2 Statistical Significance	71
4.4.3.3 Analysis of Results	74
4.4.4 Handling Observational Uncertainty	78
4.4.5 Runtime Results	85
Chapter 5: MATCH Algorithm	87
5.1 MATCH Algorithm	89
5.2 Experiment Purpose, Design, and Results	95
5.2.1 Purpose of this Study	95
5.2.2 Experimental Design	96
5.2.2.1 Participants	97
5.2.2.2 Reward Structure	98
5.2.2.3 Experimental Procedure	100
5.2.3 Results for Original Structures	103
5.2.4 Results for New Reward Structures	104
5.3 Analysis	105
5.4 λ -Re-estimation	108
5.5 Runtime Results	110
Chapter 6: Security Circumvention Games	112
6.1 TSA Security Challenges	114
6.1.1 Modeling the TSA Resource Allocation Challenges	114
6.1.1.1 Defender Strategies	115
6.1.1.2 Attacker Actions	116
6.1.2 Compact Representation for Efficiency	117

6.1.2.1	Threat Modeling for TSA	117
6.1.2.2	Compact Representation	119
6.2	Evaluation	122
6.2.1	Security Policy Analysis	122
6.2.2	Runtime Analysis	124
Chapter 7:	Conclusions	127
7.1	Summary	127
7.2	Future Work	130
Bibliography		133
Appendix A:	Statistical Significance Tests	139
A.1	Statistical Significance Tests for COBRA	139
A.2	Statistical Significance Tests for MATCH	140
Appendix B:	Reward Structures	141
Appendix C:	Strategies	160
Appendix D:	Expected Rewards for COBRA Experiments	197
Appendix E:	Expected Response Percentages for COBRA Experiment	199
Appendix F:	Strategies for varying α in COBRA($\alpha,2.5$)	200
Appendix G:	Experimental Instructions	204
G.1	Material for COBRA Experiments	204
G.2	Material for MATCH Experiments	207
G.2.1	Obvious Games	207
G.3	Experiment Instructions	208

List of Figures

2.1	LAX Security	27
4.1	Game Interface	62
4.2	Single Observation	62
4.3	Average leader expected value	69
4.4	Unobserved condition - Expected average reward	80
4.5	<i>Strategy entropy</i> for varying α values	81
4.6	Average expected values for varying α under the unlimited observation condition	82
4.7	Comparing runtimes	86
5.1	Game Interface	96
5.2	1-Norm Scatter Plots	99
5.3	Original reward structures	103
5.4	Scatter Plot of Results	107
5.5	Re-estimated Reward Structures	110
5.6	Runtime results	111
6.1	Policy Analysis: Increasing resources for 10 areas with 3 security activities per area	123
6.2	X-axis: Areas, Y-axis: Runtime	126
6.3	Runtime: Increasing resources for 10 areas with 3 security activities per area	126

G.1 Game Interface	204
G.2 Single Observation	205

Abstract

Recently, game theory has been shown to be useful for reasoning about real-world security settings where security forces must protect critical assets from potential adversaries. In fact, there have been a number of deployed real-world applications of game theory for security (e.g., ARMOR at Los Angeles International Airport and IRIS for the Federal Air Marshals Service). Here, the objective is for the security force to utilize its limited resources to best defend their critical assets.

An important factor in these real-world security settings is that the adversaries involved are humans who may not behave according to the standard assumptions of game-theoretic models. There are two key shortcomings of the approaches currently employed in these recent applications. First, human adversaries may not make the predicted rational decision. In such situations, where the security force has optimized against a perfectly rational opponent, a deviation by the human adversary can lead to adverse affects on the security force's predicted outcome. Second, human adversaries are naturally creative and security domains are highly dynamic, making enumeration of all potential threats a practically impossible task and solving the resulting game, with current leading approaches, would be intractable.

My thesis contributes to a very new area that combines algorithmic and experimental game-theory. Indeed, it examines a critical problem in applying game-theoretic techniques to situations

where perfectly rational solvers must address human adversaries. In doing so it advances the study and reach of game theory to domains where software agents and humans may interact. More specifically, to address the first shortcoming, my thesis presents two separate algorithms to address potential deviations from the predicted rational decision by human adversaries. Experimental results, from a simulation that is motivated by a real-world security domain at Los Angeles International airport, demonstrated that both of my approaches outperform the currently deployed optimal algorithms which utilize standard game-theoretic assumptions and additional alternative algorithms against humans. In fact, one of my approaches is currently under evaluation in a real-world application to aid in resource allocation decisions for the United States Coast Guard.

Towards addressing the second shortcoming of enumeration of a large number of potential adversary threat capabilities, I introduce a new game-theoretic model for efficiency, which additionally generalizes the previously accepted model for security domains. This new game-theoretic model for addressing human threat capabilities has seen real-world deployment and is under evaluation to aid the United States Transportation Security Administration in their resource allocation challenges.

Chapter 1: Introduction

Security agencies are tasked with the important challenge of protecting critical infrastructure nationwide. For instance, security agencies are required to protect transportation networks, including the passengers, from potential terrorist activities or other disruptive activities that may halt the transportation of goods and people[DHS, 2012b]. Security agencies also protect critical environmental assets. For example, agencies may be responsible for protecting national parks and wildlife from illegal animal poaching or forest extraction[GTI, 2012; EPA, 2012]. In urban neighborhoods these agencies are responsible for protecting citizens by preventing criminal activity[LAPD, 2012]. Additionally, security agencies are responsible for stopping illegal flows of drugs, weapons, and more from entering or exiting national borders along with protecting legal assets from being seized through piracy or other means[DHS, 2012a]. While there are many more important security scenarios, the common problem among them is that security agencies only have limited resources to provide this critical protection.

1.1 Problem Addressed

Stackelberg games, which were first introduced to model leadership and commitment [von Stackelberg, 1934], have become popular as an approach to address critical security scenarios similar

to those previously mentioned. Recently, a game-theoretic approach based on Stackelberg games has been successfully used to model the security challenge of optimally allocating limited security resources across a set of potential targets [Paruchuri et al., 2008; Conitzer and Sandholm, 2006; Basilico et al., 2009]. While I provide a formal definition for Stackelberg games in Section 2.1, they are a natural model for such security problems because they model the commitment a defender (e.g., security agency) must make in allocating her resources before an attacker can conduct surveillance and choose his best attack strategy, *considering the action chosen by the defender*¹.

Indeed, this approach has been featured in multiple deployed real-world applications. Two prominent ones include a deployed application for randomizing checkpoints and canine patrols at the Los Angeles International airport since August of 2007 known as ARMOR and a deployed application for randomizing Federal Air Marshals on flights since 2009 known as IRIS [Jain et al., 2010]. This approach is also being investigated for numerous other applications such as GUARDS for randomizing security activities at airports for the Transportation Security Administration [Pita et al., 2011], TRUSTS for randomizing urban security in transit systems Yin et al. [2012b], and PROTECT for randomizing port security for the United States Coast Guard [Shieh et al., 2012]. Beyond the deployed real-world applications, Stackelberg games have been used to study security problems ranging from “police and robbers” scenarios [Gatti, 2008], to computer network security [Lye and Wing, 2005], to missile defense systems [Brown et al., 2005] and terrorism [Sandler and M., 2003].

The development of these applications has inspired theoretical and algorithmic progress in efficiently applying a game-theoretic framework based on Stackelberg games to real-world security

¹By convention, I refer to the defender (leader) as *she* and attacker (follower) as *he*

scenarios. While they efficiently model strategic interactions between a defender and potential attacker, one possible failing of current models used in these deployed security applications [Jain et al., 2010; Yin et al., 2012b; Pita et al., 2011] is that they make strict assumptions on the underlying games². In real-world domains, the strict assumptions made in the underlying models may lead to significant degradation in performance. In my work I examine three of the strict assumptions made in standard game-theoretic models against human adversaries. Namely, that the human adversary is perfectly rational, that they perfectly observe the leader's strategy, and that their action space is tractable.

- *Rationality Assumption:* One of the strict assumptions made by standard game-theoretic approaches based on Stackelberg games is that the adversary is perfectly rational [Jain et al., 2010; Paruchuri et al., 2008; Conitzer and Sandholm, 2006]. That is, the adversary maximizes his expected value based on the information available. However, human adversaries may not be expected-value-maximizers, computing optimal decisions. Instead, their decisions may be governed by their bounded rationality [Simon, 1956], which causes them to deviate from their expected optimal strategy. Such unexpected deviations can have arbitrarily negative impacts on the security force's expected value if not specifically accounted for.
- *Observability Assumption:* In standard Stackelberg games it is assumed that the adversary perfectly observes the security force's strategy. In real-world settings, humans may have limited observability of the security force's strategy, giving them a false impression of that

²One exception is the application deployed since 2012 for the United States Coast Guard [Shieh et al., 2012], which relaxes these strict assumptions by utilizing an algorithm, MATCH, that I will present in this thesis to randomize patrols.

strategy. Furthermore, given limited information, humans can be biased in their decision making causing them to deviate from the optimal response of the observed strategy.

- *Action Space Assumption:* Game-theoretic models require defined (i.e., fixed) potential actions on behalf of all the players involved in order to compute an optimal mixed strategy for the security force. However, real-world security settings are dynamic in nature and humans are highly adaptable and creative allowing for the action space of an attacker to constantly be evolving. Furthermore, even if the action space was held static, the sheer number of potential attack methods by human adversaries may be too large to enumerate and solve in practice. Thus, intelligent methods for compactly representing the potential threat capabilities of human adversaries are required.

In summary, given the standard strict assumptions, humans may deviate from the predicted optimal response, which can lead to significant losses on behalf of the security force. These losses are two-fold since the security force has not optimized against this sub-optimal play, but also since the security force could have made significant gains by exploiting potential human biases. In addition, enumerating all their threat capabilities may be intractable making a solution, in general, practically infeasible to compute. Accounting for these potential deviations and large action capabilities can provide significant benefits to a security force by providing solutions that help prevent large potential losses from unpredicted behavior and exploit human biases.

1.2 Contributions

To address these three fundamental assumptions in dealing with humans my thesis contributes to a novel area of research that combines algorithmic and experimental game theory. My work

focuses on the very new and growing real-world problem of designing game-theoretic approaches to address human adversaries. Such human adversaries may not behave according to the strict assumptions made in standard game-theoretic models and thus may deviate from the predicted behavior or create complex problems that cannot be solved effectively. Specifically, I have made contributions in two key areas. The first key area of contribution is in developing algorithms to address potential deviations by human adversaries due to their bounded rationality or limited observability. These algorithms help address the first two strict assumptions presented and are presented as COBRA and MATCH. The second key area of contribution is in developing an alternative concept for addressing a human's exceedingly large and evolving action space within a game-theoretical model for efficiency. This new method for representing such a massive number of threat capabilities on behalf of the human adversary helps address the third strict assumption and I refer to this model as Security Circumvention Games.

1.2.1 COBRA/MATCH

My new algorithms aid in addressing two strict assumptions made by optimal Stackelberg solvers that an adversary is both perfectly rational and that he perfectly observes the defender strategy. These algorithms combine key ideas from: (i) previous best known algorithms for solving Bayesian Stackelberg games [Paruchuri et al., 2008], (ii) robustness approaches for games from robust optimization literature [Aghassi and Bertsimas, 2006; Ordóñez and Stier-Moses, 2007], and (iii) anchoring theories on human perception of probability distributions from psychology [Fox and Clemen, 2005]. While the robustness approaches help to address human response imprecision, anchoring, which is an expansion of support theory [Tversky and Koehler, 1994], helps address limited observational capabilities. In addition to its critical algorithmic component, my

work also differs from other behavioral based approaches in that I only consider potential deviations on behalf of the adversary while computationally assisting the security force in making rational decisions. Other behavioral approaches consider potential deviations on behalf of all players involved [Camerer, 2003; Selten, 1988; McKelvey and Palfrey, 1995]. The ideas presented in these algorithms are generalizable and in fact one of the robustness approaches I present has been used in the work of Yang et al. [2011] for an algorithm known as RPT.

This thesis introduces a mixed-integer linear program (MILP) COBRA (Combined Observability and Rationality Assumption), that builds on a previous Bayesian Stackelberg game solver known as DOBSS, which uses the standard strict assumptions [Paruchuri et al., 2008]. This MILP continues to handle adversary reward uncertainty by utilizing Bayesian Stackelberg games, however, it also addresses the uncertainty that may arise from human imprecision in choosing the expected value maximizing strategy due to bounded rationality and limited observations. As a result of this bounded rationality, the adversary may select an ϵ -optimal response strategy, i.e., the adversary may choose any of the responses within ϵ -reward of his optimal strategy. This choice may be caused by a variety of reasons, but COBRA attempts to guard against the choices that fall within this ϵ -bound of the optimal response. More specifically, given multiple possible ϵ -optimal responses, the robust approach is to assume that the attacker could choose the one that provides the security force the worst reward – not necessarily because the adversary attends to the security force’s reward, but to robustly guard against the worst-case outcome. In an adversarial setting, handling the worst case outcome may be in the best interest of the security force. This worst-case assumption contrasts with those of other Stackelberg solvers which assume the adversary will play a strong Stackelberg equilibrium (choosing a strategy that favors the leader in the case of

a tie) [Conitzer and Sandholm, 2006; Paruchuri et al., 2008], making COBRA novel to address human adversaries.

The COBRA algorithm additionally utilizes the idea of anchoring biases [Fox and Clemen, 2005; Fox and Rottenstreich, 2003; See et al., 2006b], which are based upon support theory [Tversky and Koehler, 1994], to protect against limited observation conditions when addressing human adversaries. An anchoring bias is when, given no information about the occurrence of a discrete set of events, humans will tend to assign an equal weight to the occurrence of each event (a uniform distribution). This is also referred to as giving full support to the ignorance prior [Fox and Rottenstreich, 2003]. It has been shown through extensive experimentation that humans are particularly susceptible to giving full support to the ignorance prior before they are given any information and that, once given information, they are slow to update away from this assumption [Fox and Rottenstreich, 2003]. Thus, COBRA computes a mixed strategy for the defender considering how the human will perceive this strategy based on his bias toward the ignorance prior.

While the COBRA algorithm demonstrates how accounting for potential deviations can improve the performance of defenders against human adversaries, I also present the MATCH algorithm which evolves the robustness approach of COBRA. Similar to COBRA, the MATCH algorithm avoids the complex task of explicitly modeling human decision making, which can be a difficult task in the absence of substantial human-decision-making data. MATCH leverages and modifies standard robust optimization techniques [Aghassi and Bertsimas, 2006] to create a new type of graduated optimization. Here, the defender's loss for a potential deviation by the adversary is bounded by the distance of that deviation from the expected-value-maximizing strategy. That is, instead of attempting to predict exactly how a human decision maker will deviate,

MATCH limits the degradation of the defender’s reward based on how far the human adversary deviates. In doing so, MATCH addresses the potential loss to the defender for unpredicted deviations, but still attempts to optimize against rational play. In the MATCH formulation we can control how much the degradation is limited for deviations providing a trade off between robustness to deviations and the optimal expected value of the defender for rational play by the adversary. It is important to note that, unlike COBRA, MATCH is not designed to address observational uncertainty and it is left for future work to investigate whether adding human anchoring biases into the MATCH formulation would outperform COBRA under limited observation.

In order to evaluate the usefulness of these algorithms against human adversaries, I conducted extensive experiments with human subjects playing a security game motivated by a real-world security scenario at Los Angeles International airport (LAX). The human subjects played as an attacker who had to choose 1 of 8 possible gates to attack that were guarded by three guards. The experimental results demonstrate that both COBRA and MATCH outperform optimal algorithms based on the standard game-theoretic assumptions. Furthermore, my analysis reveals that MATCH is able to outperform the current best performing algorithm for addressing human adversaries under no observational uncertainty [Yang et al., 2011], establishing the benefits of such an approach when sufficient models of human decision making are not available.

1.2.2 Security Circumvention Games

In this thesis I present an alternative concept for addressing a human’s exceedingly large and evolving action space within a game-theoretic model for efficiency. I developed a model for security settings called Security Circumvention Games (SCGs) that utilizes this new concept for defining attacker actions. Before the development of SCGs, all previously deployed work [Jain

et al., 2010] was based on a game-theoretic model known as security games [Korzhyk et al., 2011], which only allow for a single security activity and a single threat scenario. While it was efficient for the domains considered, it is not realistic in many other real-world domains. SCGs extend security games in a significant way, making the following contributions: (i) the ability to reason over heterogeneous security activities for a single target, and (ii) the ability to reason about an intelligent attacker who chooses threats designed to circumvent specific security activities. By allowing attackers circumvention capabilities we no longer need to consider specific threats, but rather which security measures they will attempt to avoid in executing a threat. By representing threats as circumvention strategies, SCGs are able to represent large-scale games in a compact way that can be solved by current Stackelberg solution techniques Paruchuri et al. [2008]; Pita et al. [2011].

In order to evaluate the value of the SCG model I examined two important issues. The first issue is with scalability and runtimes. In order for my new model to be useful it needed to be able to solve problems on the scale of real-world problems. The second issue is evaluating the value of the security policies generated using the assumptions in SCGs against potential alternative approaches. Thus, I examine SCGs against some potential standard alternative approaches for the security domains SCGs are designed for.

By addressing these fundamental challenges it is possible to improve resource allocation in real-world security settings against human adversaries. In fact, my research has been applied in the real-world. Indeed, SCGs form the core of GUARDS, a scheduling assistant under evaluation by the United States Transportation Security Administration (TSA) since Fall 2010 to schedule limited security resources within airports nationwide [Pita et al., 2011]. Furthermore, MATCH

has been deployed in the PROTECT [Shieh et al., 2012] application to assist the United States Coast Guard with randomization of their patrolling routes since Spring 2012.

1.3 Guide to Thesis

This thesis is organized in the following way. Chapter 2 introduces necessary background for the research presented in this thesis and Chapter 3 presents related work. Chapter 4 presents the algorithm COBRA and corresponding experimental results. Chapter 5 presents the algorithm MATCH and corresponding experimental results. Chapter 6 examines Security Circumvention Games, describing the model framework and justification, and compact representations for efficiency. Finally, Chapter 7 concludes the thesis and presents issues for future work.

Chapter 2: Background

The work in this thesis builds on Stackelberg games for modeling security domains. In Section 2.1 I will define a Stackelberg game for security domains and in Section 2.2 I will generalize it to a Bayesian Stackelberg game for security domains. In Section 2.3 I will describe the standard solution concept known as a strong Stackelberg equilibrium and in Section 2.4 I will introduce DOBSS, a general Stackelberg solver [Paruchuri et al., 2008] and the foundation for my COBRA and MATCH algorithms, along with two baseline algorithms (UNIFORM and MAXIMIN) for experimental comparison, which could be used in a Stackelberg setting. In Section 2.5 I will introduce the leading competitor to my approaches known as BRQR. In Section 2.6 I will define a constrained version of Stackelberg games known as security games. Finally in Section 2.7 I will describe a real-world security problem at Los Angeles International Airport (LAX) that motivates the experimental setup in this thesis and in Section 2.8 I will justify our choice of human subjects for these experiments.

2.1 Stakelberg Games

In a Stackelberg game, a leader commits to a strategy first, and then followers sequentially selfishly optimize their rewards, *considering the action chosen by the leader*. For the remainder of

this thesis I will refer to the leader as 'her' and the follower as 'him'. To see the advantage of being the leader in a Stackelberg game, consider a simple game with the payoff table as shown in Table 2.1, which was first presented by Conitzer and Sandholm [2006]. The leader is the row player and the follower is the column player. The only pure-strategy Nash equilibrium for this game is when the leader plays a and the follower plays c , which gives the leader a payoff of 2; in fact, for the leader, playing b is strictly dominated. However, if the leader can commit to playing b before the follower chooses his strategy, then the leader will obtain a payoff of 3, since the follower would then play d to ensure a higher payoff for himself. If the leader commits to a uniform mixed strategy of playing a and b with equal (0.5) probability, then the follower will play d , leading to a payoff for the leader of 3.5.

	c	d
a	2,1	4,0
b	1,0	3,2

Table 2.1: Payoff table for example Stackelberg game.

The Stackelberg games I consider in this thesis have two agents, the leader (defender), Θ , and the follower (attacker/adversary), Ψ . Each player has a set of possible *pure strategies*, denoted $\Sigma_{\Theta} = \{\theta_1, \dots, \theta_{|\Sigma_{\Theta}|}\}$ for the leader and $\Sigma_{\Psi} = \{\psi_1, \dots, \psi_{|\Sigma_{\Psi}|}\}$ for the follower. A *mixed strategy* allows a player to play a probability distribution over pure strategies, denoted $x \in X$ for the leader and $q \in Q$ for the follower. Here, $x_i \in [0, 1]$ represents the probability with which the leader takes pure strategy $\theta_i \in \Sigma_{\Theta}$ and similarly $q_i \in [0, 1]$ represents the probability with which the follower takes pure strategy $\psi_i \in \Sigma_{\Psi}$. I denote by S_{Θ} the index set over the leader's pure strategies and

by S_Ψ the index set over the follower's pure strategies. Payoffs are respectively defined for the leader and the follower over all possible joint pure strategy outcomes:

$$\Omega_\Theta : \Sigma_\Theta \times \Sigma_\Psi \rightarrow \mathbb{R} \quad (2.1)$$

$$\Omega_\Psi : \Sigma_\Theta \times \Sigma_\Psi \rightarrow \mathbb{R} \quad (2.2)$$

The payoff functions are extended to mixed strategies in the standard way, by taking the expectation over pure-strategy outcomes. That is, given a leader strategy $x \in X$ and a follower strategy $q \in Q$ the leader's and follower's payoffs can be determined by:

$$\Omega_\Theta(x, q) = \sum_{i \in S_\Theta} \sum_{j \in S_\Psi} \Omega_\Theta(\theta_i, \psi_j) \cdot x_i \cdot q_j \quad (2.3)$$

$$\Omega_\Psi(x, q) = \sum_{i \in S_\Theta} \sum_{j \in S_\Psi} \Omega_\Psi(\theta_i, \psi_j) \cdot x_i \cdot q_j \quad (2.4)$$

2.2 Bayesian Stackelberg Games

The Bayesian extension to Stackelberg games allows for each player (leader or follower) to be one of multiple possible types, with each type associated with its own payoff values. In this thesis the defender (leader), Θ , only has one type since she is considering her own personal resources. However, the attacker (follower) can be one of a set of possible types denoted by $\gamma \in \Gamma$. For example, a security force may be interested in protecting against potential terrorist attacks and catching potential drug smugglers, which represent two different types of adversaries. Each type is represented by a different and possibly uncorrelated payoff matrix for both the leader and follower. That is, the leader's payoffs will vary along with the follower's payoffs for each type of

follower. At any time the leader does not know what follower type she will face, however, she is aware of the probability distribution over follower types (i.e., she knows how frequently she will face each follower type). The probability with which follower type $\gamma \in \Gamma$ appears is denoted by p^γ . The follower is always aware of his own type and thus always has perfect information about the leader's payoffs and his own payoffs.

While the set of possible pure strategies remains the same for both the leader and the follower (Σ_Θ and Σ_Ψ respectively), payoffs for each player are now defined over all possible pure strategy outcomes for each follower type:

$$\Omega_\Theta : \Sigma_\Theta \times \Sigma_\Psi \times \Gamma \rightarrow \mathbb{R} \quad (2.5)$$

$$\Omega_\Psi : \Sigma_\Theta \times \Sigma_\Psi \times \Gamma \rightarrow \mathbb{R} \quad (2.6)$$

Additionally, each follower type $\gamma \in \Gamma$ can choose his own mixed strategy denoted by $q^\gamma \in Q$. For a particular follower type $\gamma \in \Gamma$, the payoff functions are still extended to mixed strategies in the standard way, by taking the expectation over pure-strategy outcomes. That is, given a follower type $\gamma \in \Gamma$, a leader strategy $x \in X$ and a follower strategy $q^\gamma \in Q$ the leader's and follower's payoffs can be determined by:

$$\Omega_\Theta(x, q^\gamma, \gamma) = \sum_{i \in S_\Theta} \sum_{j \in S_\Psi} \Omega_\Theta(\theta_i, \psi_j, \gamma) \cdot x_i \cdot q_j^\gamma \quad (2.7)$$

$$\Omega_\Psi(x, q^\gamma, \gamma) = \sum_{i \in S_\Theta} \sum_{j \in S_\Psi} \Omega_\Psi(\theta_i, \psi_j, \gamma) \cdot x_i \cdot q_j^\gamma \quad (2.8)$$

Given this formal model, the leader’s goal is to determine the mixed strategy $x \in X$, such that her expected value is maximized given that each follower type will choose his expected-value-maximizing action with complete knowledge of the leader’s mixed strategy. Thus, the follower’s strategy in a Stackelberg game is a function that selects a strategy in response to each leader strategy for each follower type:

$$F_{\Psi} : X \times \Gamma \rightarrow Q \tag{2.9}$$

Such a commitment to a mixed strategy models a real-world situation where security forces commit to a randomized patrolling strategy first. Given this commitment, an adversary can conduct as much surveillance of this mixed strategy as he desires. Even with knowledge of this mixed strategy, the adversary has no specific knowledge of what the security force may do on a particular day however. He only has knowledge of the mixed strategy the security force will use to decide her resource allocations for that day. In this model, predictable defense strategies are vulnerable to exploitation by a determined adversary.

2.3 Strong Stackelberg Equilibrium

Leitmann [1978] proposed two types of unique Stackelberg equilibria, which are typically called “strong” and “weak” after Breton et al. [1988]. The strong form assumes that the follower will always choose the optimal strategy for the leader in cases of indifference, while the weak form assumes that the follower will choose the worst strategy for the leader in cases of indifference. A strong Stackelberg equilibrium exists in all Stackelberg games, but a weak Stackelberg equilibrium may not [Basar and Olsder, 1995]. In addition, the leader can often induce the favorable

strong equilibrium by selecting a strategy arbitrarily close to the equilibrium that causes the follower to strictly prefer the desired strategy [von Stengel and Zamir, 2004]. Recent research utilizing a Stackelberg framework with the standard assumptions of rationality and perfect observability for security domains has adopted strong Stackelberg equilibrium [Kiekintveld et al., 2009; Jain et al., 2011b].

Definition 1. *A pair of strategies (x, F_Ψ) form a Strong Stackelberg Equilibrium if they satisfy the following:*

1. *The leader plays a best response:*

$$\sum_{\gamma \in \Gamma} \Omega_\Theta(x, F_\Psi(x, \gamma), \gamma) \geq \sum_{\gamma \in \Gamma} \Omega_\Theta(x', F_\Psi(x', \gamma), \gamma) \quad \forall x' \in X$$

2. *The follower plays a best-response:*

$$\sum_{\gamma \in \Gamma} \Omega_\Psi(x, F_\Psi(x, \gamma), \gamma) \geq \sum_{\gamma \in \Gamma} \Omega_\Psi(x, q, \gamma) \quad \forall x \in X, q \in Q$$

3. *The follower breaks ties optimally for the leader:*

$$\sum_{\gamma \in \Gamma} \Omega_\Theta(x, F_\Psi(x, \gamma), \gamma) \geq \sum_{\gamma \in \Gamma} \Omega_\Theta(x, q, \gamma) \quad \forall x \in X, q \in Q^*(x), \text{ where } Q^*(x) \text{ is the set of follower best-responses to } x, \text{ as above.}$$

2.4 DOBSS and Baseline Algorithms

For completeness this thesis includes both a uniformly random strategy and a MAXIMIN strategy against human opponents as a baseline against the performance of both existing algorithms, such as DOBSS, and the new algorithms presented. Proposed algorithms must outperform the two baseline algorithms to provide benefits.

2.4.1 DOBSS

I now describe the Decomposed Optimal Bayesian Stackelberg Solver (DOBSS) in detail as it provides a starting point for COBRA. While the problem of choosing an optimal strategy for the leader in a Stackelberg game is NP-hard for a Bayesian game with multiple follower types [Conitzer and Sandholm, 2006], researchers have continued to provide practical improvements. DOBSS is an efficient general Stackelberg solver [Paruchuri et al., 2008] and is in use for security scheduling at the Los Angeles International Airport. It operates directly on the compact Bayesian representation, giving speedups over the multiple linear programs method [Conitzer and Sandholm, 2006], which requires conversion of the Bayesian game into a normal-form game by the Harsanyi transformation [Harsanyi and Selten, 1972]. In particular, DOBSS obtains a decomposition scheme by exploiting the property that follower types are independent of each other. The key to the DOBSS decomposition is the observation that evaluating the leader strategy against a Harsanyi-transformed game matrix is equivalent to evaluating against each of the game matrices for the individual follower types and then obtaining a weighted sum.

I first present DOBSS in its most intuitive form as a Mixed-Integer Quadratic Program (MIQP); I then present a linearized equivalent Mixed-Integer Linear Program (MILP). The DOBSS model explicitly represents the actions by the leader and the *optimal* actions for the follower types in the problem solved by the leader. Note that DOBSS needs to consider only the expected-value-maximizing pure strategies of the follower types, since for a given fixed mixed strategy $x \in X$ of the leader, each follower type, $\gamma \in \Gamma$, faces a problem with fixed linear rewards. If a mixed strategy is optimal for the follower, then so are all the pure strategies in support of that mixed strategy.

In the MIQP presented below, let M be a large positive number. Given prior probabilities p^γ , with $\gamma \in \Gamma$, of facing each follower type, the leader solves the following:

$$\begin{aligned} \max \quad & \sum_{i \in S_\Theta} \sum_{j \in S_\Psi} \sum_{\gamma \in \Gamma} p^\gamma \cdot \Omega_\Theta(\theta_i, \psi_j, \gamma) \cdot x_i \cdot q_j^\gamma \\ \text{s.t.} \quad & \sum_{i \in S_\Theta} x_i = 1 \end{aligned} \quad (2.10)$$

$$x_i \in [0 \dots 1] \quad \forall i \in S_\Theta \quad (2.11)$$

$$\sum_{j \in S_\Psi} q_j^\gamma = 1 \quad \forall \gamma \in \Gamma \quad (2.12)$$

$$q_j^\gamma \in \{0, 1\} \quad \forall \gamma \in \Gamma, j \in S_\Psi \quad (2.13)$$

$$0 \leq a^\gamma - \sum_{i \in S_\Theta} \Omega_\Psi(\theta_i, \psi_j, \gamma) \cdot x_i \leq (1 - q_j^\gamma) \cdot M \quad \forall \gamma \in \Gamma, j \in S_\Psi \quad (2.14)$$

$$a^\gamma \in \mathbb{R} \quad \forall \gamma \in \Gamma \quad (2.15)$$

Here for a leader strategy x and a strategy q^γ for each follower type, the objective represents the expected reward for the leader considering the *a priori* distribution over different follower types p^γ . Constraints 2.10 and 2.11 define the set of feasible solutions $x \in X$ as a probability distribution over the set of pure strategies $\theta_i \in \Sigma_\Theta$. Constraints 2.12 and 2.13 limit the mixed strategies of follower type γ , $q^\gamma \in Q$, to be only pure strategies over the set Σ_Ψ (that is each q^γ has exactly one coordinate equal to one and the rest equal to zero). The two inequalities in Constraint 2.14 ensure that $q_j^\gamma = 1$ only for a strategy j that is optimal (i.e, expected-value-maximizing) for follower type γ . Indeed this is a linearized form of the optimality conditions for the linear programming problem solved by each follower type. Constraint 2.14 can be explained as follows:

the leftmost inequality ensures that $\forall \gamma \in \Gamma, j \in S_\Psi : a^\gamma \geq \sum_{i \in S_\Theta} \Omega_\Psi(\theta_i, \psi_j, \gamma) \cdot x_i$. This means that given the leader's strategy x , a^γ is an upper bound on follower type γ 's reward for any strategy. The rightmost inequality is inactive for every strategy where $q_j^\gamma = 0$, since M is a large positive quantity. For the strategy that has $q_j^\gamma = 1$ this inequality states $a^\gamma \leq \sum_{i \in S_\Theta} \Omega_\Psi(\theta_i, \psi_j, \gamma) \cdot x_i$, which combined with the previous inequality shows that this strategy must be optimal for follower type γ .

This quadratic programming problem can be linearized through the change of variables $z_{ij}^\gamma = x_i \cdot q_j^\gamma$, thus obtaining the mixed integer linear programming problem presented below. My implementation of DOBSS solves this MILP, which was shown to be equivalent to the quadratic programming problem presented above and the equivalent Harsanyi transformed Stackelberg game by Paruchuri et al. [2008]. The DOBSS MILP can be solved using efficient integer programming packages. For a more in depth explanation of DOBSS please see Paruchuri et al. [2008].

While DOBSS is an efficient general Stackelberg solver, it has three fundamental problems in the mixed strategies it computes given a human adversary (follower). First, DOBSS assumes the human adversary will break ties in the defender's (leader's) favor. However, in real-world security scenarios, which are adversarial in nature, it is plausible that a human adversary would not make any differentiation between ties or possibly even break ties against the defender leading to a reduced defender reward. Second, DOBSS assumes the human adversary is perfectly rational and thus will choose an expected-value-maximizing strategy among all of his alternatives. However, if the human adversary is boundedly rational and deviates from an expected-value-maximizing strategy it can severely impact the defender's expected value. Finally, in real-world security scenarios it may be difficult for a human adversary to perfectly observe the defender's strategy.

Such imperfect observation can lead the human adversary to deviate from the expected-value-maximizing strategy, which can again have adverse affects on the defender's expected value.

This thesis helps address these critical limitations in the DOBSS formulation.

$$\begin{aligned} \max \sum_{i \in S_\Theta} \sum_{j \in S_\Psi} \sum_{\gamma \in \Gamma} p^\gamma \cdot \Omega_\Theta(\theta_i, \psi_j, \gamma) \cdot z_{ij}^\gamma \\ \text{s.t. } \sum_{i \in S_\Theta} \sum_{j \in S_\Psi} z_{ij}^\gamma = 1 \quad \forall \gamma \in \Gamma \end{aligned} \quad (2.16)$$

$$\sum_{j \in S_\Psi} z_{ij}^\gamma \leq 1 \quad \forall \gamma \in \Gamma, i \in S_\Theta \quad (2.17)$$

$$z_{ij}^\gamma \in [0 \dots 1] \quad \forall \gamma \in \Gamma, i \in S_\Theta, j \in S_\Psi \quad (2.18)$$

$$q_j^\gamma \leq \sum_{i \in S_\Theta} z_{ij}^\gamma \leq 1 \quad \forall \gamma \in \Gamma, j \in S_\Psi \quad (2.19)$$

$$\sum_{j \in S_\Psi} q_j^\gamma = 1 \quad \forall \gamma \in \Gamma \quad (2.20)$$

$$q_j^\gamma \in \{0, 1\} \quad \forall \gamma \in \Gamma, j \in S_\Psi \quad (2.21)$$

$$0 \leq a^\gamma - \sum_{i \in S_\Theta} \Omega_\Psi(\theta_i, \psi_j, \gamma) \left(\sum_{h \in S_\Psi} z_{ih}^\gamma \right) \leq (1 - q_j^\gamma) \cdot M \quad \forall \gamma \in \Gamma, j \in S_\Psi \quad (2.22)$$

$$\sum_{j \in S_\Psi} z_{ij}^\gamma = \sum_{j \in S_\Psi} z_{ij}^1 \quad \forall \gamma \in \Gamma, i \in S_\Theta \quad (2.23)$$

$$a^\gamma \in \mathbb{R} \quad \forall \gamma \in \Gamma \quad (2.24)$$

2.4.2 UNIFORM

UNIFORM is the most basic method of randomization which just assigns an equal probability of taking each strategy $\theta_i \in \Sigma_\Theta$ (a uniform distribution).

2.4.3 MAXIMIN

MAXIMIN is a traditional approach which assumes the follower may take any of the available actions. The objective of the following linear program is to maximize the minimum reward, V , the leader will obtain irrespective of the follower's action:

$$\begin{aligned} \max \sum_{\gamma \in \Gamma} p^\gamma \cdot V^\gamma \\ \text{s.t. } \sum_{i \in S_\Theta} x_i = 1 \end{aligned} \tag{2.25}$$

$$\sum_{i \in S_\Theta} \Omega_\Theta(\theta_i, \psi_j, \gamma) \cdot x_i \geq V^\gamma \quad \forall \gamma \in \Gamma, j \in S_\Psi \tag{2.26}$$

$$x_i \in [0 \dots 1] \quad \forall i \in S_\Theta \tag{2.27}$$

2.5 BRQR

Best Response to Quantal Response (BRQR) is based on the quantal response (QR) model [McKelvey and Palfrey, 1995] of human decision making. This QR model is a well-founded solution concept in game theory derived from Nobel-prize-winning work in choice modeling theory [Nobelprize.org, 2012], and there has been significant support for this QR model elsewhere in the literature [Wright and Leyton-Brown, 2010]. The QR model suggests that instead of strictly maximizing utility, individuals respond stochastically in games: the chance of selecting a non-optimal strategy increases as the cost of such an error decreases. Specifically, the QR model defines a parameter, λ , which represents the amount of noise in a player's response.

In applying the QR model, BRQR only adds noise to the response function for the adversary, so the defender computes an optimal strategy assuming the attacker responds with a noisy best-response. Given λ and the defender's mixed-strategy x , the adversaries' quantal response q_j (i.e., the probability of taking pure strategy $\psi_j \in \Sigma_\Psi$) can be written as:

$$q_j = \frac{e^{\lambda \sum_{i \in S_\Theta} \Omega_\Psi(\theta_i, \psi_j) \cdot x_i}}{\sum_{k \in S_\Psi} e^{\lambda \sum_{i \in S_\Theta} \Omega_\Psi(\theta_i, \psi_k) \cdot x_i}} \quad (2.28)$$

The goal is to maximize the defender's expected utility given q (i.e., $\sum_{i \in S_\Theta} \sum_{j \in S_\Psi} \Omega_\Theta(\theta_i, \psi_j) \cdot x_i \cdot q_j$). Combined with Equation 2.28, the problem of finding the optimal mixed strategy for the defender can be formulated as:

$$\begin{aligned} \max \sum_{i \in S_\Theta} \sum_{j \in S_\Psi} \Omega_\Theta(\theta_i, \psi_j) \cdot x_i \cdot \frac{e^{\lambda \sum_{i \in S_\Theta} \Omega_\Psi(\theta_i, \psi_j) \cdot x_i}}{\sum_{k \in S_\Psi} e^{\lambda \sum_{i \in S_\Theta} \Omega_\Psi(\theta_i, \psi_k) \cdot x_i}} \\ \text{s.t. } \sum_{i \in S_\Theta} x_i = 1 \end{aligned} \quad (2.29)$$

$$x_i \in [0 \dots 1] \quad \forall i \in S_\Theta \quad (2.30)$$

A major difficulty of this modeling approach however, is that it requires the appropriate estimation of λ to determine the level of noise in the human adversary's response function. Yang et al. [2011] have proposed a method for appropriately estimating λ within security settings. Still, in real-world scenarios that are often complex and large in scope, creating an accurate model of human decision making or estimating such parameters can be a difficult task. This difficulty is exacerbated in security settings where information about potential adversaries is often sparse and noisy.

2.6 Security Stackelberg Games

At this time I introduce a restricted version of a Stackelberg game known as a security game [Korzhyk et al., 2011]. The MATCH algorithm that I will present in Chapter 5 is designed specifically for this restricted class of games because in a general Stackelberg framework a solution to the MATCH algorithm may not exist. Security games make strict assumptions on the payoffs of both the leader and the follower in a Stackelberg game with security settings specifically in mind. In security games, there are a set of *targets* $T = \{t_1, \dots, t_{|T|}\}$ that the defender wants to protect from an attacker and a set of *resources* $R = \{r_1, \dots, r_{|R|}\}$ (e.g., police officers) that the defender may deploy to protect the targets. Resources are identical in that any resource can be deployed to protect any target, and provide equivalent protection to that target. Furthermore, there is no additional benefit to deploying more than one resource to any individual target (i.e., once a target is protected by a resource it is considered fully protected).

The payoffs for the defender and attacker depend on which target is attacked, and whether the target is protected (covered) or not (uncovered). Formally, the defender's and attacker's payoffs are defined for each target depending on whether it is covered (c) or uncovered (u):

$$U_{\Theta}^u : T \rightarrow \mathbb{R} \quad (2.31)$$

$$U_{\Theta}^c : T \rightarrow \mathbb{R} \quad (2.32)$$

$$U_{\Psi}^u : T \rightarrow \mathbb{R} \quad (2.33)$$

$$U_{\Psi}^c : T \rightarrow \mathbb{R} \quad (2.34)$$

Let Equation 2.35 denote the difference between the defender's covered and uncovered payoffs. Similarly, Equation 2.36 denotes the difference between the attacker's uncovered and covered payoffs. As a key property of security games, it is necessary that $\Delta U_{\Theta}(t_i) > 0$ and $\Delta U_{\Psi}(t_i) > 0$. In other words, adding resources to cover a target helps the defender and hurts the attacker.

$$\Delta U_{\Theta}(t_i) = U_{\Theta}^c(t_i) - U_{\Theta}^u(t_i) \quad \forall t_i \in T \quad (2.35)$$

$$\Delta U_{\Psi}(t_i) = U_{\Psi}^u(t_i) - U_{\Psi}^c(t_i) \quad \forall t_i \in T \quad (2.36)$$

For the purpose of this thesis I only consider security games that have schedules of size 1 (i.e., satisfy the Subsets of Schedules Are Schedules property [Korzhyk et al., 2011]). A pure strategy of the defender then, still denoted by $\theta_i \in \Sigma_{\Theta}$, is a subset of targets from T with size equal to $|R|$. An adversary's pure strategy, still denoted by $\psi_j \in \Sigma_{\Psi}$, is exactly one target $t_j \in T$. Given that I only consider security games with schedules of size 1, Kiekintveld et al. [2009] have shown that the defender's possible mixed strategies can be transformed into an equivalent coverage vectors of probability, $c \in C$, where c_i is the marginal probability of covering t_i and similarly a coverage vector can be transformed into an equivalent valid mixed strategy $x \in X$ of the defender [Korzhyk et al., 2011].

Given that payoffs for both the defender and attacker only depend on whether a particular target $t_i \in T$ is covered or not, the marginal probability $c \in C$ can be used in place of the mixed strategy $x \in X$ to compute the defender's optimal resource allocation, reducing the complexity of the defender's actions. Specifically, there are $\binom{|T|}{|R|}$ potential pure strategies and only $|T|$ marginal probabilities. Where before $\sum_{i \in S_{\Theta}} x_i = 1$ for a defender's mixed strategy, it is now the case that $\sum_{i \in T} c_i = |R|$ and $c_i \in [0 \dots 1] \quad \forall i \in T$ for the defender's coverage vector. In other words, the

defender wants to use all of her resources and placing more than 1 resource on a target (i.e., more than 100% coverage) is not beneficial. Here, given a coverage vector $c \in C$ and an attacker mixed strategy $q \in Q$, the defender's and attacker's payoffs are defined in Equations 2.37 and 2.38 respectively. Additionally, as Kiekintveld et al. [2009] have previously defined, the *attack set*, $\pi(c)$, contains all targets that yield the maximum expected value for the attacker given coverage c as shown in Equation 2.39. In a SSE, the attacker selects the target in the attack set with the maximum expected value for the defender, which I denote by t^* . It follows that the expected SSE value for the defender is $\hat{U}_\Theta(c) = U_\Theta(c, t^*)$ and for the attacker $\hat{U}_\Psi(c) = U_\Psi(c, t^*)$.

$$U_\Theta(c, q) = \sum_{i \in S_\Psi} q_i (c_i \cdot U_\Theta^c(t_i) + (1 - c_i) U_\Theta^u(t_i)) \quad (2.37)$$

$$U_\Psi(c, q) = \sum_{i \in S_\Psi} q_i (c_i \cdot U_\Psi^c(t_i) + (1 - c_i) U_\Psi^u(t_i)) \quad (2.38)$$

$$\pi(c) = \{t : U_\Psi(c, t) \geq U_\Psi(c, t') \quad \forall t' \in T\} \quad (2.39)$$

While COBRA builds off the general Stackelberg solver DOBSS, MATCH leverages the unique structure of security games to provide improved runtime benefits in computing the defender's resource allocation. MATCH is able to reason over the marginal coverage vector, greatly reducing the defender's action space from $\binom{T}{R}$ potential actions to $|T|$ potential actions. In doing so, MATCH is able to solve for larger scale problems, which is necessary in the deployed PROTECT application for the US Coast Guard [Shieh et al., 2012] that uses the MATCH algorithm.

2.7 Los Angeles International Airport

While there are a number of applications that utilize a Stackelberg framework [Jain et al., 2010; Shieh et al., 2012; Pita et al., 2011], I chose to model my experimental setup after the security scenario at Los Angeles International Airport because it is the least constrained and simplest real-world deployed system making it ideal for an initial investigation against human subjects. I describe the specific challenges in the security problems faced by the Los Angeles World Airport (LAWA) police. Los Angeles International Airport (LAX) is the fifth busiest airport in the United States, the largest destination airport in the United States, and serves 60-70 million passengers per year [LAWA, 2012; Stevens et al., 2006]. LAX is unfortunately also suspected to be a prime terrorist target on the west coast of the United States, with multiple arrests of plotters attempting to attack LAX [Stevens et al., 2006]. To protect LAX, LAWA police have designed a security system that utilizes multiple rings of protection. As is evident to anyone traveling through LAX airport, these rings include such things as vehicular checkpoints, police units patrolling the roads to the terminals and inside the terminals (with canines), and security screening and bag checks for passengers. There are unfortunately not enough resources (police officers) to monitor every single event at the airport; given its size and number of passengers served, such a level of screening would require considerably more personnel and cause greater delays to travelers. Thus, assuming that all checkpoints and terminals are not being monitored at all times, setting up available checkpoints, canine units or other patrols on deterministic schedules allows adversaries to learn the schedules and plot an attack that avoids the police checkpoints and patrols, which makes deterministic schedules ineffective.



(a) LAX Checkpoint



(b) Canine Patrol

Figure 2.1: LAX Security

Randomization offers a solution here. In particular, from among all the security measures that randomization could be applied to, LAWA police have so far posed two crucial problems for the deployed ARMOR system [Pita et al., 2008]. First, given that there are many roads leading into LAX, where and when they should set up checkpoints to check cars driving into LAX. For example, Figure 2.1(a) shows a vehicular checkpoint set up on a road inbound towards LAX. Police officers examine cars that drive by, and if any car appears suspicious, they do a more detailed inspection of that car. LAWA police wished to obtain a randomized schedule for such checkpoints for a particular time frame. For example, if they are to set up two checkpoints, and the time frame of interest is 8 AM to 11 AM, then a candidate schedule may suggest to the police that on Monday, checkpoints should be placed on route 1 and route 2, whereas on Tuesday during the same time slot, they should be on route 1 and 3, and so on. Second, LAWA police wished to obtain an assignment of canines to patrol routes through the terminals inside LAX. For example, if there are three canine units available, a possible assignment may be to place canines on terminals 1, 3, and 6 on the first day, but terminal 2, 4, and 6 on another day and so on based on the available information. Figure 2.1(b) illustrates a canine unit on patrol at LAX.

Given these problems, the ARMOR system considers three key challenges [Pita et al., 2008]: (i) potential attackers can observe security forces' schedules over time and then choose their attack strategy – the fact that the adversary acts with knowledge of the security forces' schedule makes deterministic schedules highly susceptible to attack; (ii) there is unknown and uncertain information regarding the types of adversary LAX police may face; (iii) although randomization helps eliminate deterministic patterns, it must also account for the different costs and benefits associated with particular targets.

I use this real-world security problem and domain as the basis for my experimental setup. In particular I examine the canine patrolling problem of LAX where there are 8 terminals and 3 canines patrolling those terminals (although at LAX there may be more or less canines on patrol at any time). However, beyond the key challenges the ARMOR system addressed [Pita et al., 2008], I examine two critical assumptions of the underlying DOBSS algorithm in use for the ARMOR system [Pita et al., 2008]. Namely, that the human adversary is perfectly rational and perfectly observes the defender strategy.

2.8 Human Subjects

Since my algorithms are centered on addressing non-optimal and uncertain human responses, traditional proofs of correctness and optimality are insufficient: it is necessary to experimentally test these new approaches against existing approaches. Experimental analysis with human subjects allows me to show how my algorithms are expected to perform against human adversaries compared to alternative approaches. To that end, I perform my experiments against a general

population of undergraduate and graduate students in engineering at the University of Southern California for COBRA and across a random demographic of people residing in the United States for MATCH. Given that the experimental results for COBRA and MATCH are across a random demographic of people, it is necessary to justify that these approaches are useful against terrorists specifically. This is particularly important since MATCH is already deployed for the United States Coast Guard. While it is virtually impossible to test my approaches against other approaches with actual terrorists who will behave according to their normal decision-making processes, I will argue that the experiments using students and United States residents provide results in the right direction.

The key question is if the terrorist psychiatric profile may lead to vastly different decision-making processes than the general population. An argument could be made that terrorists are completely irrational or suffer from severe psychosis causing them to make near arbitrary decisions. Thus they do not attempt to maximize expected-value or even attempt to make boundedly rational decisions making any game-theoretic approach, even one that considers potential deviations due to bounded rationality, irrelevant. However, research has shown that the primary shared characteristic of the psychiatric profile of the decision makers of terrorist factions is their normalcy [Abrahms, 2008; Richardson, 2006; Gill and Young, 2011]. This extensive research suggests that terrorists are actually highly rational and carefully construct their attack strategies attempting to maximize their expected value, justifying a game-theoretic approach [Rosoff and John, 2009; Keeney and von Winterfeldt, 2010; Abrahms, 2008; Richardson, 2006]. As Louise Richardson, who has done significant research on the psyche of terrorists, states “But terrorists, by and large, are not insane at all. Their primary shared characteristic is their normalcy, insofar as we understand the term. Psychological studies of *terrorists* are virtually unanimous on this

point” [Richardson, 2006]. Furthermore, Abrahms [2008] both agrees on this point and notes that these studies are based on records of dozens of terrorist organizations from the late 1960s to the present. It follows that testing against the general population can give some strategic insights on how to better defend against potential terrorists attacks given the inability to test against actual terrorists.

Based on the assumption that terrorists are rational agents, an argument could be made that it should be sufficient to calculate a game-theoretic optimal resource allocation strategy. However, modern decision theory recognizes that human decision makers face cognitive and informational constraints. Additionally, terrorists sometimes face competing objectives and noisy information [Allison and Zelikow, 1999; Abrahms, 2008]. Thus, as my algorithms purpose, deviations are likely due to bounded rationality and cognitive limitations.

While the algorithms presented in this thesis are designed with the real-world applications of Stackelberg games for preventing terrorist based activities specifically in mind, another benefit of experimenting with the general population is that the approaches presented in this thesis can be applied to a large number of criminal scenarios beyond terrorism. Criminal activities can be broadly broken down into six categories: (i) Property crimes, (ii) violent crimes, (iii) sex crimes, (iv) gangs and crime, (v) white-collar occupational crime, and (vi) drugs and crime [Pogrebin, 2012]. It is the responsibility of many different security agencies to attempt to prevent these criminal activities, with each agency focusing on different categories of crime. The approaches presented in this thesis can potentially aid a number of these agencies. For instance, urban police may need to patrol a large number of neighborhoods in an attempt to prevent property crimes. If the police take a deterministic approach, visiting each neighborhood in a set sequence, then a determined criminal could monitor their approach and exploit this pattern to potentially rob an

innocent victim. As another example, the Internal Revenue Service (IRS) may audit a company for fraudulent activities if they notice strange activity. However, the IRS could also use this approach to audit additional companies on a randomized basis in an attempt to catch any activity they may have missed. Given the large range of criminal based activities, the resulting types of human criminals will span a wide variety making the use of a general population largely applicable.

In the future, this type of approach may be further refined for specific types of criminals or criminal activities. The idea is that a personality and demographic profile can be defined for criminals who participate in a particular type of criminal activity and we can experimentally evaluate the decision-making process for people of that profile. In doing so, we could tailor an approach to that particular decision-making process, thus improving the benefits. For example, research has shown that terrorists tend to be risk-averse and attempt to avoid uncertainty in planning their attack strategies [Morral and Jackson, 2009]. By studying risk-averse subjects it may be possible to improve upon the approaches presented in this thesis for the deployed applications discussed previously [Jain et al., 2010; Shieh et al., 2012]. However, whenever performing experimental evaluation of different approaches it can be particularly difficult to obtain the correct population that needs to be examined and this difficulty has been considered before in other behavioral game-theoretic studies [Camerer, 2003].

Chapter 3: Related Work

The related work on game-theoretic approaches for security can be broadly divided into three categories: (i) efficient solutions for computing optimal Stackelberg equilibria, (ii) computing *robust* strategies, and (iii) addressing deviations from the theoretically optimal choice.

3.1 Computing Optimal Stackelberg Equilibria

Numerous algorithms have been proposed for computing strong Stackelberg equilibria (SSE) strategies for Bayesian Stackelberg games. These can be further broken down into two categories: (i) algorithms for computing efficient solutions to general Bayesian Stackelberg games, and (ii) algorithms for computing efficient solutions for large-scale security games (large defender and attacker action spaces). This related work largely complements the work presented in this thesis since it is possible to combine the efficient approaches presented in this related work with my new approaches for addressing human adversaries.

3.1.1 Efficient Solutions to general Bayesian Stackelberg games

Computing a solution to a Bayesian Stackelberg game has been previously shown to be an NP-hard problem, making scale-up a difficult challenge [Conitzer and Sandholm, 2006]. One of the

first optimal Bayesian Stackelberg solvers is the *multiple-LPs* approach, which solves many linear programs to compute the optimal defender strategy in Bayesian Stackelberg games [Conitzer and Sandholm, 2006]. Building on this approach, DOBSS [Paruchuri et al., 2008] decomposes the Bayesian game into individual types and solves a mixed-integer linear program to compute the strong Stackelberg equilibrium (SSE) strategy. In Section 2.4 I include an in depth examination of the DOBSS algorithm as it forms the foundation for the COBRA algorithm I will present. Following in this direction, HBGS uses a novel technique based on hierarchical decomposition and branch and bound search over the follower type space and has been shown to be orders of magnitude faster than previous approaches [Jain et al., 2011b]. Similarly, Jain et al. [2011b] present a new exact algorithm called HBSA, which extends the previous fastest known security game solver towards the Bayesian case. While COBRA and MATCH do not utilize the approaches presented in HBGS and HBSA, it is possible that in the future these approaches could be applied to improve the runtime performance of both algorithms.

The presented approaches address Bayesian Stackelberg games with a finite number of follower types. However, work has also been done to address infinite or continuous Bayesian Stackelberg games allowing for payoffs to be represented using continuous payoff distributions as opposed to discrete payoffs. Kiekintveld et al. [2011] present several techniques for finding approximate solutions for this class of games, and show empirically that these approaches demonstrate improvements over the alternative approaches presented. Kiekintveld et al. [2011] present SBE, SRD, GMC, and DTS as approaches for approximating solutions to such games and empirically evaluate the effectiveness of each against previous best known solvers. Building both upon the work of finite Bayesian Stackelberg games and infinite Bayesian Stackelberg games, Yin and Tambe [2012] present the HUNTER algorithm. The HUNTER algorithm was designed to both

address a discrete number of attacker types and, based on infinite Bayesian Stackelberg games, continuous uncertainty such as the follower's observation noise, the leader's execution error, and both players' payoff uncertainty.

By considering observation noise, execution uncertainty, and payoff uncertainty; HUNTER provides a type of robustness that is similar to COBRA and MATCH. However, there are some key differences in the approach taken by HUNTER and that taken by COBRA and MATCH. HUNTER attempts to maximize the mean outcome of a known distribution of error or noise by using sample average approximation theories as opposed to maximizing the worst-case outcome. Furthermore, HUNTER still considers a perfectly rational adversary who will make the optimal choice given that error or noise in observation, execution, and/or payoff. HUNTER does not consider that a human adversary may deviate even further due to bounded rationality or biases such as an anchoring bias given observation uncertainty. COBRA and MATCH consider specific types of deviations from human adversaries versus arbitrary deviations.

Assuming there is no uncertainty, HUNTER's approach may help in addressing human adversaries who may deviate due to bounded rationality by artificially introducing some error. However, this type of approach considers the mean of arbitrary deviations within that error level versus the directed deviations of COBRA and MATCH. In the future, combining the general robustness approach of HUNTER and the directed robustness of COBRA and MATCH may provide further benefits in domains where observation noise, execution uncertainty, and/or payoff uncertainty are present even beyond the cognitive limitations of the human adversary.

3.1.2 Efficient Solutions for Large-Scale Security Games

ORIGAMI and ERASER-C are algorithms that have been developed for larger and more complex security settings that exploit the underlying structure of security games specifically [Kiekintveld et al., 2009]. ORIGAMI is a polynomial time algorithm that computes optimal security force strategies for a security game when no scheduling constraints are present by exploiting the constraints on the reward structure of the game. The ERASER-C mixed-integer linear program allows for certain kinds of resource and scheduling constraints. Formally, the set of legal schedules $S = \{s_1 \dots s_l\}$ is a subset of the power set of the targets, with restrictions on this set representing scheduling constraints. ERASER-C provides scale-ups by computing the security force coverage per schedule, instead of computing mixed strategies over a joint assignment for all security force resources. Unfortunately, Kiekintveld et al. [2009] show that ERASER-C may fail to generate a correct solution in cases where the security force may face arbitrary scheduling constraints.

Improving upon this work, Jain et al. [2010] present ASPEN, which utilizes a novel branch-and-price approach to overcome both the scaling challenge and the challenge of generating correct solutions given arbitrary scheduling constraints. Here, the algorithm starts by considering a minimal set of pure strategies for both the players (security force and adversary) and then iteratively generate additional pure strategies that will improve the payoff of the corresponding player (e.g., a security force's pure strategy is added if it helps increase the security force's payoff). This process is repeated until the optimal solution is obtained. ASPEN has empirically been shown to both be competitive with ERASER-C for the restricted class of games where ERASER-C is applicable and to solve far more general instances of the scheduling problems where ERASER-C and other existing techniques fail.

Beyond the class of games where resources are assigned to static targets, there are also numerous patrolling problems and corresponding efficient solution techniques within the security game setting. Basilio et al. [2009] present an efficient game-theoretic approach utilizing a Stackelberg framework for optimal patrolling of arbitrary topologies. Similarly, Agmon et al. [2011] present an approach for patrolling a closed perimeter or open polyline (fence) to minimize an adversaries chance of penetrating the perimeter or open polyline. Johnson et al. [2012] examine a different patrolling problem where the goal is to maximize the radius of a successful patrol against would be forest extractors in order to preserve the largest pristine forest area. Similarly, Yin et al. [2012b] present an initial model and corresponding linear program for computing optimal patrol strategies in urban transit systems with certain temporal and spacial constraints.

Finally, there exists a class of algorithms for scheduling resources on static targets within a graph based or network structure. Similar to the other work described, the goal here is to maximize the security forces overall reward given that the adversary will attempt to find their way from some starting node to some target destinations with differing value. RANGER is a heuristic linear program for solving this complex problem, which efficiently creates optimal marginal distributions for placing checkpoints [Tsai et al., 2010]. However, it is proven that the reward found by RANGER is an overestimate of the true optimal reward. Jain et al. [2011a] have shown that RANGER can be arbitrarily bad in general settings and have presented RUGGED, which is the first scalable optimal solution technique for such network security games. RUGGED, like ASPEN, utilizes a column generation approach where it starts by considering a minimal set of pure strategies for both players.

3.2 Computing Robust Strategies

This line of research is most similar to that presented in this thesis, however, a crucial difference is that this work considers robustness against a rational adversary in the face of certain types of uncertainty whereas I consider the case where the security force deals with a boundedly rational adversary who may deviate even without uncertainty in the game. Additionally, I consider the *human* adversary's reaction to certain types of uncertainty.

Korzhyk et al. [2011] consider the limiting case where an attacker has no observations and thus investigate the equivalence of Stackelberg vs Nash equilibria. Yin et al. [2012a] develop RECON to compute robust solutions in the case where there may be noise in the security force's execution of the suggested mixed strategy and/or the observations made by an adversary can be noisy. RECON attempts to maximize the worst case outcome within some execution and/or observation noise or error level from the intended strategy. Similarly, An et al. [2012] consider the condition when the adversary does not have perfect surveillance capabilities, and thus are unable to learn the exact strategy of the defender. A new model is presented, security games with strategic surveillance (SGSS), which models the adversary's belief update and strategic surveillance decisions in security games. In addition, An et al. [2012] provide multiple formulations for computing the defender's optimal strategies, including non-convex programming and convex approximation, and provide an approximate approach for computing the adversary's optimal surveillance length.

Kiekintveld et al. [2011] model distributions over preferences of an adversary using infinite Bayesian games, and propose a number of algorithms (SBE, SRD, GMC, and DTS) to generate approximate solutions for such games. A critical assumption of traditional game-theoretic

solvers is that both the security force and adversary are perfectly aware of all the payoffs involved. Here, the concept is to be robust to potential uncertainty in the adversary's target payoffs. Building both upon the work for observation/execution uncertainty and payoff uncertainty, Yin et al. [2012b] present the HUNTER algorithm. The HUNTER algorithm was designed to provide a unified method for creating robust strategies against the adversary's observation noise, the security force's execution error, and both players' payoff uncertainty. Again, here HUNTER attempts to maximize the mean outcome within a given level of uncertainty.

Beyond execution, observation, and payoff uncertainty, secrecy and deception have also been modeled for Stackelberg games [Zhuang and Bier, 2011]. Outside of Stackelberg games, models for execution uncertainty in game-theory have been separately developed [Archibald and Shoham, 2009]. Additionally, robust solution methods for simultaneous move games have been studied [Aghassi and Bertsimas, 2006; Porter et al., 2002].

3.3 Addressing Suboptimal Decisions

A common concern in game-theoretic settings has been considering potential deviations from the game-theoretic optimal play [Camerer, 2003; Selten, 1988; McKelvey and Palfrey, 1995]. One well known approach is trembling hand perfect equilibrium [Selten, 1988] where players find an equilibrium given that the other players may choose unintended strategies due to a "trembling hand" or error. A trembling hand perfect equilibrium is a sequence of perturbed games (games where only totally mixed strategies are allowed) that converge to some base game Nash equilibria. The idea here is that even if the second player trembles, it will not give the first player incentive

to deviate from the current Nash equilibrium and vice versa. Another approach is that of ϵ -equilibrium [Tijs, 1981] where it is assumed that in simultaneous move games an ϵ -equilibrium point has been reached when a unilateral deviation from that equilibrium point by one of the players will not increase the payoff of that player by more than ϵ .

In addition to these key equilibrium concepts, the field of experimental game theory has provided a wealth of contributions and insights on deviations of human play from equilibrium predictions and different models to explain these deviations [Camerer, 2003]. In contrast with these equilibrium concepts and human models, there are three key differences in the work in my thesis: (i) the equilibrium approaches seek stable equilibrium points where the players involved are not expected to deviate and thus do not specifically address the impact of unexpected deviations or protect against them. In contrast, my approaches robustly guard against potential deviations due to sub-optimal play; (ii) both the equilibrium concepts and human models consider strategies where all players involved may deviate whereas my approach computes a rational strategy for the defender (i.e., an approach that the defender is expected to follow without deviation) by utilizing insights on the potential deviations of a boundedly rational human opponent; and (iii) the equilibrium approaches only define stable equilibrium whereas my work also considers the computational aspects of finding an optimal strategy.

More similar to the work presented in this thesis, there has been significant work in developing agents that consider interactions with humans [Haim et al., 2012; Azaria et al., 2012; de Melo et al., 2011]. Here, the agents' goal is to improve the humans performance or decision making on some task based on the agents' interaction with that human. For instance, Azaria et al. [2012] consider the design of an automated advice provision agent who will have repeated interactions with a human user. Similarly, de Melo et al. [2011] consider the effect of agents with emotional

expression when negotiating with humans. Marsella et al. [2010] consider the role and history of computational models of human emotion in general and their wide variety of applications. A key difference is that this work considers the design of agents who directly interact with humans and attempt to improve the outcome of these interactions whereas I consider adversarial settings where an agent plans against a human adversary, but will not have any direct interaction with that human adversary in order to influence his decisions.

Particularly in line with my work, Yang et al. [2011] have examined incorporating better models of human decision making within the Stackelberg framework for addressing human adversaries in security games. Specifically, Yang et al. [2011] have designed algorithms to compute solutions based on prospect theory [Kahneman and Tversky, 1979] and the quantal response (QR) model [McKelvey and Palfrey, 1995] to address human adversaries. Prospect theory describes human decision making as a process of maximizing 'prospect' rather than maximizing expected value (i.e., humans do not decide according to a risk neutral utility function). Here, Yang et al. [2011] compute an optimal strategy for the security force considering that the adversary will respond according to the prospect theory model. The quantal response (QR) model [McKelvey and Palfrey, 1995] of human decision making is a well-founded solution concept in game theory derived from Nobel-prize-winning work in choice modeling theory [Nobelprize.org, 2012]. The QR model suggests that instead of strictly maximizing expected value, individuals respond stochastically in games: the chance of selecting non-optimal strategies increases as the cost of such an error decreases. In applying the QR model to security games, noise is only added to the response function for the adversary, so the security force computes an optimal strategy assuming the attacker responds with a noisy best-response. However, this approach critically depends on

the appropriate estimation of λ , which represents the amount of error or noise in the attacker's response function.

The key difference between the work of Yang et al. [2011] and the work presented in this thesis is that Yang et al. [2011] focus on finding appropriate models of human decision making while my work focuses on finding robust strategies for humans in the absence of such models. In general, having a perfect model of human-decision-making processes would lead to the optimal strategy for security forces against human adversaries. However, finding such perfect models is a difficult task and often requires a large amount of human-decision-making data to create, which can be limited in security settings. Additionally, McCubbins et al. [2012] have recently shown that the currently existing models of human decision making are fundamentally flawed. In the absence of such data and strong models of human decision making, the robust approaches I propose in this thesis are a beneficial substitute and perform significantly better than currently existing approaches for addressing human adversaries.

Chapter 4: COBRA Algorithm

As previously discussed, there exists a number of algorithms for Bayesian Stackelberg games that find optimal solutions considering an *a priori* probability distribution over possible follower types [Conitzer and Sandholm, 2006; Paruchuri et al., 2008; Kiekintveld et al., 2009; Jain et al., 2011b, 2010; Yin and Tambe, 2012]. Unfortunately, to guarantee optimality, these particular algorithms make strict assumptions on the underlying games, namely that the players are perfectly rational and that the followers perfectly observe the leader's strategy. However, these assumptions rarely hold in real-world domains, particularly when dealing with humans. Of specific interest are security domains where there are limited resources to protect a critical set of targets [Jain et al., 2010; Pita et al., 2011; Shieh et al., 2012] – even though an automated program may determine an optimal leader (defender) strategy, it must take into account a human follower (adversary).

Such human adversaries may not be expected-value maximizers, computing optimal decisions. Instead, their decisions may be governed by their bounded rationality [Simon, 1956] which causes them to deviate from their expected optimal strategy. Humans may also suffer from limited observability of the defender's strategy, given them a false impression of that strategy. In other words, when making decisions based on their own cognitive abilities, humans are biased due to their bounded rationality and inability to obtain complete sets of observations. Thus, a human

adversary may not respond with the game theoretic optimal choice, causing the defender to face uncertainty over the gamut of adversary's actions.

Therefore, in this work, the leader in a Stackelberg game must commit to a strategy considering three different types of uncertainty: (i) adversary response uncertainty due to his bounded rationality where the adversary may not choose the expected-value-maximizing strategy, (ii) adversary response uncertainty due to his limitations in appropriately observing the leader strategy, and (iii) adversary reward uncertainty modeled as different reward matrices with a Bayesian *a priori* distribution assumption (i.e., a Bayesian Stackelberg game). While game-theoretic optimal algorithms such as DOBSS, which are based on the standard strict game-theoretic assumptions, can handle the third type of uncertainty, these models can give a severely under-performing strategy when the adversary deviates because of the first two types of uncertainty. This degradation in leader rewards may be unacceptable in certain domains.

To remedy this situation, I draw inspiration from robust optimization methodology, in which the decision maker optimizes against the worst outcome over some uncertainty [Aghassi and Bertsimas, 2006; Nilim and Ghaoui, 2005]. I also draw inspiration from psychological support theory for human decision making when humans are given a discrete set of actions and an unknown probability function over those actions [See et al., 2006b; Tversky and Koehler, 1994]. In the presented Stackelberg problem, the leader will make a robust decision by considering that the follower, who may not follow expected-value-maximizing rationality, could choose a strategy from his range of possible responses that degrades the leader expected value the most or that he could choose a strategy that is based on his limited observations.

To that end, I introduce a mixed-integer linear program (MILP), COBRA (Combined Observability and Rationality Assumptions), that builds on the Bayesian Stackelberg game model

in DOBSS. This MILP continues to handle adversary reward uncertainty in the same fashion as DOBSS. Along with handling reward uncertainty, it also addresses the uncertainty that may arise from human imprecision in choosing the expected optimal strategy due to bounded rationality and limited observations. Namely, it introduces the idea of robust responses to ϵ -optimal follower responses into DOBSS and Stackelberg games in general. It also utilizes the concept of anchoring biases to protect against limited observation conditions, handling observational uncertainty. I first describe in depth the key ideas behind COBRA and then incrementally define the MILP that uses them.

4.1 Key Ideas

The two main ideas in COBRA are addressing boundedly rational opponents and anchoring biases those opponents may have.

4.1.1 Bounded Rationality

COBRA assumes that the follower is boundedly rational and may not strictly maximize expected value. As a result, the follower may select an ϵ -optimal response strategy, i.e., the follower may choose any of the responses within ϵ -expected-value of his optimal strategy. This choice may be caused by a variety of reasons, but COBRA attempts to guard against the choices that fall within this ϵ -bound of the optimal response.

More specifically, given multiple possible ϵ -optimal responses, the robust approach is to assume that the follower could choose the one that provides the leader the worst expected value – not necessarily because the follower attends to the leader reward, but to robustly guard against

the worst-case outcome. In an adversarial setting, handling the worst-case outcome may be in the best interest of the leader. This worst-case assumption contrasts those of other Stackelberg solvers which assume the follower will play a strong Stackelberg equilibrium (choosing a strategy that favors the leader in the case of a tie) [Conitzer and Sandholm, 2006; Paruchuri et al., 2008], making COBRA novel to address human followers.

4.1.2 Anchoring Theory

Anchoring is a cognitive bias where humans rely too heavily (i.e., anchor) on a trait or piece of information when making decisions. Specifically, once a human has anchored on some piece of information, their decision making is biased toward adjusting or interpreting other information to reflect the anchored information. Tversky and Koehler [1994] developed a theory of subjective probability known as support theory, which examines one particular type of anchoring bias where subjects anchor on the way information is presented to them. Research based on support theory has shown that when, given no information about the occurrence of a discrete set of events, humans will tend to assign an equal weight to the occurrence of each event (a uniform distribution) [Fox and Clemen, 2005; Fox and Rottenstreich, 2003; See et al., 2006b]. This is also referred to as giving full support to the ignorance prior where the ignorance prior represents a subjects baseline belief (i.e., in this case the uniform distribution over events) [Fox and Rottenstreich, 2003].

It has been shown through extensive experimentation that humans are particularly susceptible to giving full support to this ignorance prior (uniform distribution) before they are given any information and that, once given information, they are slow to update away from this assumption [Fox and Clemen, 2005; Fox and Rottenstreich, 2003; See et al., 2006b]. Thus they leave some

support, $\alpha \in [0 \dots 1]$, on the ignorance prior and the rest, $1 - \alpha$, on the occurrence they have actually viewed. As humans become more confident in what they are viewing, this bias begins to diminish, decreasing the value of α .

Models have been proposed to address this bias and predict what probability a human will assign to a particular event e from a set of events E based on the evaluative assessment (i.e., assessment based on events actually viewed) they have made for the occurrence of that event. Let e represent a particular event, $E \setminus e$ represent the remaining events possible, let $P(e), P(E \setminus e)$ be the real probabilities and $P(e'), P((E \setminus e)')$ represent the probability a human assigns to event e and $E \setminus e$ respectively. One model [Fox and Rottenstreich, 2003] defines the human estimated probabilities with the following ratios: $P(e')/P((E \setminus e)') = (|e|/|E \setminus e|)^\alpha \cdot (P(e)/P(E \setminus e))^{1-\alpha}$. Here, $|e|/|E \setminus e|$ is the ratio in the case of uniform probabilities and represents the ignorance prior. The α value indicates the relative contribution of these two sources of information. Note that as α approaches 1, the estimated probability converges on the uniform assumption, while when α approaches 0, it is closer to the true probability distribution. Research suggests that as people gain more relevant knowledge they will give less support to the ignorance prior and more support to evaluative assessment thus decreasing the value of α [Fox and Clemen, 2005; Fox and Rottenstreich, 2003; See et al., 2006b].

An alternative model assumes estimated probabilities are directly calculated using a simple linear model [Fox and Clemen, 2005; Fox and Rottenstreich, 2003; Tversky and Koehler, 1994]: $P(e') = \alpha \cdot (1/|E|) + (1 - \alpha) \cdot P(e)$. I commandeer this anchoring bias for Stackelberg games to determine how a human follower may perceive the leader strategy. For example, in the game shown in Table 2.1, suppose the leader strategy was to play a with a probability of 0.8 and b with 0.2. Anchoring bias would predict that in the absence of any information ($\alpha = 1$), humans will

assign a probability of 0.5 to each of a and b , and will only update this belief (alter the value of α) after observing the leader strategy for some time. Although these may not be the only possible models for determining anchoring bias, they are standard in the related literature [Fox and Rottenstreich, 2003; Tversky and Koehler, 1994] and the linear model is ideal since the odds form model is not easily representable in an MILP.

As an alternative approach I could use Bayesian updating to predict how humans will perceive the probability of each event from a set of events after obtaining some observations. However, there is more support in the literature that humans act according to subjective probability and anchoring biases rather than performing Bayesian updating when evaluating evidence [Fox and Clemen, 2005; Fox and Rottenstreich, 2003; Kahneman and Tversky, 1972; See et al., 2006b; Tversky and Koehler, 1994]. Also, using Bayesian updates requires tracking which observations the humans have received, while in the real world defenders will not be aware of which days adversaries take observations and which days they do not. Anchoring bias is more general in that it allows the defender to work with just an estimate of how many observations she believes an adversary will take rather than which specific observations he takes. Specifically, a value is assigned to α based on how much evidence the defender thinks the human will receive. If the human adversary is expected to observe the defender's policy frequently and carefully then α will be low while if the defender suspects that the adversary will not have many observations of her policy, α will be high.

4.2 Robust Algorithm

$\text{COBRA}(\alpha, \epsilon)$ is the new algorithm I introduce in this thesis. To introduce it in steps, I will first introduce two simplified versions, which I will refer to as $\text{COBRA}(0, \epsilon)$ and $\text{COBRA}(\alpha, 0)$. $\text{COBRA}(0, \epsilon)$ deals only with bounded rationality and $\text{COBRA}(\alpha, 0)$ deals only with observational uncertainty. After introducing each of these pieces individually I will combine them into a single algorithm that can handle both types of uncertainty which I refer to as $\text{COBRA}(\alpha, \epsilon)$. For each of these algorithms α and ϵ represent two parameters that can be adjusted.

4.2.1 $\text{COBRA}(0, \epsilon)$

$\text{COBRA}(0, \epsilon)$ considers the case of a boundedly-rational follower, where $\text{COBRA}(0, \epsilon)$ maximizes the minimum expected value it obtains from any ϵ -optimal response from the follower. In the following MILP, I use the same variable notation as in DOBSS from Chapter 2 Section 2.4:

$$\begin{aligned} & \max \sum_{\gamma \in \Gamma} p^\gamma \cdot V^\gamma \\ & \text{s.t. } \sum_{i \in S_\Theta} x_i = 1 \end{aligned} \quad (4.1)$$

$$x_i \in [0 \dots 1] \quad \forall i \in S_\Theta \quad (4.2)$$

$$\sum_{j \in S_\Psi} q_j^\gamma = 1 \quad \forall \gamma \in \Gamma \quad (4.3)$$

$$\sum_{j \in S_\Psi} h_j^\gamma \geq 1 \quad \forall \gamma \in \Gamma \quad (4.4)$$

$$q_j^\gamma \leq h_j^\gamma \quad \forall \gamma \in \Gamma, j \in S_\Psi \quad (4.5)$$

$$q_j^\gamma, h_j^\gamma \in \{0, 1\} \quad \forall \gamma \in \Gamma, j \in S_\Psi \quad (4.6)$$

$$0 \leq a^\gamma - \sum_{i \in S_\Theta} \Omega_\Psi(\theta_i, \psi_j, \gamma) \cdot x_i \leq (1 - q_j^\gamma) \cdot M \quad \forall \gamma \in \Gamma, j \in S_\Psi \quad (4.7)$$

$$\epsilon \cdot (1 - h_j^\gamma) \leq a^\gamma - \sum_{i \in S_\Theta} \Omega_\Psi(\theta_i, \psi_j, \gamma) \cdot x_i \leq \epsilon + (1 - h_j^\gamma) \cdot M \quad \forall \gamma \in \Gamma, j \in S_\Psi \quad (4.8)$$

$$V^\gamma \leq M \cdot (1 - h_j^\gamma) + \sum_{i \in S_\Theta} \Omega_\Theta(\theta_i, \psi_j, \gamma) \cdot x_i \quad \forall \gamma \in \Gamma, j \in S_\Psi \quad (4.9)$$

$$a^\gamma \in \mathbb{R} \quad \forall \gamma \in \Gamma \quad (4.10)$$

As before, the variables q_j^γ identify the optimal strategy for follower type γ with a value of a^γ in Constraints 4.3 and 4.7. Variables h_j^γ represent all ϵ -optimal strategies for follower type γ ; Constraint 4.4 allows selection of more than one ϵ -optimal strategy per follower type. Constraint 4.8 ensures that $h_j^\gamma = 1$ for every action ψ_j such that $a^\gamma - \sum_{i \in S_\Theta} \Omega_\Psi(\theta_i, \psi_j, \gamma) \cdot x_i < \epsilon$, since in this case the middle term in the inequality is less than ϵ and the left inequality is then only satisfied if $h_j^\gamma = 1$. This robust approach required the design of a new objective and an additional constraint.

Constraint 4.9 helps define the objective value against follower type γ , V^γ , which must be lower than any leader expected value for all actions $h_j^\gamma = 1$, as opposed to the DOBSS formulation which has only one action $q_j^\gamma = 1$. Setting V^γ to the minimum leader expected value allows COBRA(0, ϵ) to robustly guard against the worst-case scenario.

4.2.2 COBRA(α , 0)

COBRA(α , 0) considers the case where the human follower is perfectly rational, but faces limited observations. COBRA(α , 0) draws upon the theory of anchoring biases mentioned previously to help address the human uncertainty that arises from such limited observation. It deals with two strategies: (i) the real leader strategy (x) and (ii) the perceived strategy by the follower (x'), where x' is defined by the linear model presented earlier. Thus, x_i is replaced in Constraint 4.15 with x'_i and x'_i is accordingly defined as $x'_i = \alpha \cdot (1 \setminus |\Sigma_\Theta|) + (1 - \alpha) \cdot x_i$.

The justification for this replacement is as follows. First, this particular constraint ensures that the follower maximizes his expected value. Since the follower believes x' to be the leader strategy then he will choose his strategy according to x' and not x . Second, given this knowledge, the leader can find the follower's responses based on x' and optimize her actual strategy x against this strategy. Since x' is a combination of the support for x and the support toward the ignorance prior, COBRA(α , 0) is able to find a strategy x that will maximize the leader's expected value based on the relative contribution of these two sources of support. For consistency among these new approaches I use the same objective introduced in COBRA(0, ϵ). The new MILP then is as follows:

$$\begin{aligned} & \max \sum_{\gamma \in \Gamma} p^\gamma \cdot V^\gamma \\ & \text{s.t. } \sum_{i \in S_\Theta} x_i = 1 \end{aligned} \quad (4.11)$$

$$x_i \in [0 \dots 1] \quad \forall i \in S_\Theta \quad (4.12)$$

$$\sum_{j \in S_\Psi} q_j^\gamma = 1 \quad \forall \gamma \in \Gamma \quad (4.13)$$

$$q_j^\gamma \in \{0, 1\} \quad \forall \gamma \in \Gamma, j \in S_\Psi \quad (4.14)$$

$$0 \leq a^\gamma - \sum_{i \in S_\Theta} \Omega_\Psi(\theta_i, \psi_j, \gamma) \cdot x'_i \leq (1 - q_j^\gamma) \cdot M \quad \forall \gamma \in \Gamma, j \in S_\Psi \quad (4.15)$$

$$V^\gamma \leq M \cdot (1 - q_j^\gamma) + \sum_{i \in S_\Theta} \Omega_\Theta(\theta_i, \psi_j, \gamma) \cdot x_i \quad \forall \gamma \in \Gamma, j \in S_\Psi \quad (4.16)$$

$$a^\gamma \in \mathbb{R} \quad \forall \gamma \in \Gamma \quad (4.17)$$

$$x'_i = \alpha \cdot (1 \setminus |\Sigma_\Theta|) + (1 - \alpha) \cdot x_i \quad \forall i \in S_\Theta \quad (4.18)$$

4.2.3 COBRA(α, ϵ)

COBRA(α, ϵ) is an MILP that combines both a bounded rationality assumption and an observational uncertainty assumption. This is achieved by incorporating the alterations made in COBRA($\alpha, 0$) and COBRA($0, \epsilon$) into a single MILP. Namely, COBRA(α, ϵ) includes both the ϵ parameter and the α parameter from COBRA($0, \epsilon$) and COBRA($\alpha, 0$) respectively. The MILP that follows is identical to COBRA($0, \epsilon$) except that in Constraints 4.24 and 4.25, x_i is replaced with x'_i as it is in COBRA($\alpha, 0$). The justification for this replacement is the same as in COBRA($\alpha, 0$).

The new MILP then is as follows:

$$\begin{aligned} \max \quad & \sum_{\gamma \in \Gamma} p^\gamma \cdot V^\gamma \\ \text{s.t.} \quad & \sum_{i \in S_\Theta} x_i = 1 \end{aligned} \quad (4.19)$$

$$x_i \in [0 \dots 1] \quad \forall i \in S_\Theta \quad (4.20)$$

$$\sum_{j \in S_\Psi} q_j^\gamma = 1 \quad \forall \gamma \in \Gamma \quad (4.21)$$

$$\sum_{j \in S_\Psi} h_j^\gamma \geq 1 \quad \forall \gamma \in \Gamma \quad (4.22)$$

$$q_j^\gamma \leq h_j^\gamma \quad \forall \gamma \in \Gamma, j \in S_\Psi \quad (4.23)$$

$$q_j^\gamma, h_j^\gamma \in \{0, 1\} \quad \forall \gamma \in \Gamma, j \in S_\Psi \quad (4.24)$$

$$0 \leq a^\gamma - \sum_{i \in S_\Theta} \Omega_\Psi(\theta_i, \psi_j, \gamma) \cdot x'_i \leq (1 - q_j^\gamma) \cdot M \quad \forall \gamma \in \Gamma, j \in S_\Psi \quad (4.25)$$

$$\epsilon \cdot (1 - h_j^\gamma) \leq a^\gamma - \sum_{i \in S_\Theta} \Omega_\Psi(\theta_i, \psi_j, \gamma) \cdot x'_i \leq \epsilon + (1 - h_j^\gamma) \cdot M \quad \forall \gamma \in \Gamma, j \in S_\Psi \quad (4.26)$$

$$V^\gamma \leq M \cdot (1 - h_j^\gamma) + \sum_{i \in S_\Theta} \Omega_\Theta(\theta_i, \psi_j, \gamma) \cdot x_i \quad \forall \gamma \in \Gamma, j \in S_\Psi \quad (4.27)$$

$$a^\gamma \in \mathbb{R} \quad \forall \gamma \in \Gamma \quad (4.28)$$

$$x'_i = \alpha \cdot (1 \setminus |\Sigma_\Theta|) + (1 - \alpha) \cdot x_i \quad \forall i \in S_\Theta \quad (4.29)$$

4.2.4 Complexity

It has been shown that finding an optimal solution in a Bayesian Stackelberg game is NP-hard [Conitzer and Sandholm, 2006] and thus DOBSS, COBRA(α , 0), COBRA(0, ϵ), and COBRA(α , ϵ) are MILPs that face an NP-hard problem. A number of effective solution packages for MILPs can be used, but their performance depends on the number of integer variables. DOBSS and

COBRA($\alpha, 0$) consider $|\Sigma_\Psi||\Gamma|$ integer variables, while COBRA($0, \epsilon$) and COBRA(α, ϵ) double that. MAXIMIN on the other hand is a linear programming problem that can be solved in polynomial time. Thus it is anticipated that MAXIMIN will have the lowest running time per problem instance, followed by DOBSS and COBRA($\alpha, 0$) with COBRA($0, \epsilon$) and COBRA(α, ϵ) close behind. However, I will show in Section 4.4.5, this was not observed in practice.

4.3 Equivalences Between Models

In this section I suggest and prove equivalences between COBRA(α, ϵ) and DOBSS under certain conditions. First I will demonstrate how DOBSS can be reformulated with the same objective as COBRA($\alpha, 0$). After this reformulation I will show how altering the parameters α and ϵ can cause COBRA(α, ϵ), and DOBSS to produce identical results (i.e., identical mixed strategies).

Observation 1. *When $\epsilon = 0$ and $\alpha = 0$ then COBRA(α, ϵ) and DOBSS are equivalent*

Proof. It follows from the definition of x'_i that when $\alpha = 0$ then $x'_i = x_i$ since the follower is assumed to once again perfectly observe and believe the leader strategy x_i . Note that if $\epsilon = 0$ the inequality in Constraint 4.25 of COBRA(α, ϵ) is the same expression as the inequality in Constraint 4.24 with h_j^γ substituted for q_j^γ . Since the objective of COBRA(α, ϵ) is to maximize the leader expected value this means that q_j^γ will be selected as the follower's optimal response that maximizes the leader expected value (i.e., a strong Stackelberg equilibrium) and h_j^γ will be set to the same. Introducing additional $h_k^\gamma = 1$ where $k \neq j$ would only serve to reduce the expected value and thus would not be an optimal solution to the MILP. I will show that DOBSS and COBRA($0, 0$) attain the same optimal objective function value.

To show that solution to COBRA(0,0) \geq solution to DOBSS, consider (q, z, a) a feasible solution for DOBSS. We define $\bar{x}_i = \sum_{j \in S_\Psi} z_{ij}^\gamma$, $\bar{q} = \bar{h} = q$, $\bar{a} = a$, and $\bar{V}^\gamma = \sum_{i \in S_\Theta} \sum_{j \in S_\Psi} \Omega_\Theta(\theta_i, \psi_j, \gamma) \cdot z_{ij}^\gamma$. From Constraints 2.7, 2.8, 2.10, and 2.14 in DOBSS we can show that $z_{ij}^\gamma = 0$ for all j such that $q_j^\gamma = 0$ and thus that $\bar{x}_i = z_{ij}^\gamma$ for all j such that $q_j^\gamma = 1$. This implies that $\bar{V}^\gamma = \sum_{i \in S_\Theta} \Omega_\Theta(\theta_i, \psi_j, \gamma) \cdot \bar{x}_i$ for the j such that $q_j^\gamma = 1$ in DOBSS and it is then easy to verify that $(\bar{x}, \bar{q}, \bar{h}, \bar{a}, \bar{V})$ is feasible for COBRA(0, 0) with the same objective function value of (q, z, a) in DOBSS.

For solution to DOBSS \geq solution to COBRA(0,0), consider (x, q, h, a, V) feasible for COBRA(0,0). Define $\bar{q} = q$, $\bar{z}_{ij}^\gamma = x_i \cdot q_j^\gamma$, and $\bar{a} = a$. Then we can show that $(\bar{q}, \bar{z}, \bar{a})$ is feasible for DOBSS by construction. For COBRA(0,0), since $q_j^\gamma \leq h_j^\gamma$ in Constraint 4.22 and as explained in the optimal solution h_j^γ will equal q_j^γ it follows that $V^\gamma \leq \sum_{i \in S_\Theta} \Omega_\Theta(\theta_i, \psi_j, \gamma) \cdot x_i$ for the j such that $q_j^\gamma = 1$. This implies that $V^\gamma \leq \sum_{i \in S_\Theta} \sum_{j \in S_\Psi} \Omega_\Theta(\theta_i, \psi_j, \gamma) \cdot \bar{z}_{ij}^\gamma$ and that the objective function value of $(\bar{q}, \bar{z}, \bar{a})$ in DOBSS is greater than or equal to the objective value of (x, q, h, a, V) in COBRA(0,0). \square

The key implication of the above observation is that when $\epsilon = 0$, COBRA(α, ϵ) loses its robustness feature, so that once again when the follower faces a tie, it selects a strategy favoring the leader, as in DOBSS. Based on this observation, the remaining observations presented in this thesis about COBRA(α, ϵ) can be generalized to DOBSS accordingly.

Observation 2. *When α is held constant, the optimal expected value COBRA(α, ϵ) can obtain is decreasing in ϵ .*

Proof. Since Constraint 4.25 in COBRA(α, ϵ) makes $h_j^\gamma = 1$ when that action has a follower expected value between $(a^\gamma - \epsilon, a^\gamma]$, increasing ϵ would increase the number of follower strategies

set to 1. Having more active follower actions in Constraint 4.26 can only decrease the minimum value of V^γ . \square

Observation 3. *Regardless of α , if $\frac{1}{2} \cdot \epsilon > P \geq |\Omega_\Psi(\theta_i, \psi_j, \gamma)| \ \forall i, j, \gamma$, where P is the greatest absolute opponent payoff, then $\text{COBRA}(\alpha, \epsilon)$ is equivalent to MAXIMIN .*

Proof. Note that $|a^\gamma|$ in $\text{COBRA}(\alpha, \epsilon) \leq P$. The proof simply needs to show that the leftmost inequality of Constraint 4.25 in $\text{COBRA}(\alpha, \epsilon)$ implies that all h_j^l must equal 1. This would make $\text{COBRA}(\alpha, \epsilon)$ equivalent to MAXIMIN . We know from the observation itself that $-P \leq \sum_{i \in S_\Theta} \Omega_\Psi(\theta_i, \psi_j, \gamma) \cdot x_i$. Suppose some $h_j^l = 0$, then we can reorganize the leftmost inequality of Constraint 4.25 to state that $\sum_{i \in S_\Theta} \Omega_\Psi(\theta_i, \psi_j, \gamma) \cdot x_i \leq a^\gamma - \epsilon$. From the observation itself we can then obtain $a^\gamma - \epsilon < P - 2P = -P$ and thus we have $-P \leq \sum_{i \in S_\Theta} \Omega_\Psi(\theta_i, \psi_j, \gamma) \cdot x_i < -P$ a contradiction. \square

Although DOBSS and $\text{COBRA}(0, \epsilon)$ make different assumptions about the follower's responses, it can be shown that in a zero-sum game their optimal solutions become equivalent. This of course is not true of general sum games. In a zero-sum game the rewards of the leader and follower are related by

$$\Omega_\Theta(\theta_i, \psi_j, \gamma) = -\Omega_\Psi(\theta_i, \psi_j, \gamma) \ \forall i \in S_\Theta, j \in S_\Psi, \gamma \in \Gamma.$$

It is well known that maximin strategies constitute the only natural solution concept for two player zero-sum games [von Neumann, 1927]. As DOBSS is an optimal Stackelberg solver it follows that it is equivalent to a maximin strategy for zero-sum games.

Observation 4. *$\text{COBRA}(0, \epsilon)$ is equivalent to a maximin strategy for zero-sum games.*

Proof. Given the maximin solution to a zero-sum game we define the expected value for any strategy of the follower, $\psi_j \in \Sigma_\Psi$, to be W_j^Θ for the leader and W_j^Ψ for the follower. Assuming that the follower's strategies are ordered in descending order in terms of expected value for the follower and there are m such strategies, the maximin solution yields $W_1^\Psi \geq W_2^\Psi \geq \dots \geq W_m^\Psi$ for the follower and $W_1^\Theta \leq W_2^\Theta \leq \dots \leq W_m^\Theta$ for the leader. COBRA(0, ϵ) is an algorithm that assumes the attacker will choose a strategy within ϵ of his maximum expected value (in this case W_1^Ψ) and of these ϵ -optimal strategies it attempts to maximize the expected value for the worst-case outcome. As seen by $W_1^\Theta \leq \dots \leq W_m^\Theta$ the minimum expected value possible for the leader is the follower's optimal strategy and any ϵ -deviation will only result in a higher expected value for the leader. It follows that the maximin strategy already maximizes the expected value for the worst-case outcome of any ϵ -optimal response by the follower. Thus, COBRA(0, ϵ) yields a strategy that is equivalent to maximin in zero-sum games. \square

4.4 Experiment Purpose, Design, and Results

4.4.1 Purpose of this Study

I sought to investigate the performance of several previously existing approaches against my new robust approach (COBRA(α , ϵ)) under several variables. In particular, I examine the performance of COBRA(α , ϵ) against DOBSS and the two baseline approaches (MAXIMIN and UNIFORM) described in Section 2.4. Here, performance is measured by the average expected value obtained by a security force against the decisions of human adversaries. These experiments were setup to examine three crucial variables of real-world domains: i) reward structure, ii) observation

condition, and iii) parameter settings for α and ϵ . In the following sections I will describe each of these variables in depth.

The goal of COBRA(α, ϵ) was to increase the performance of a security force against human adversaries by addressing the bounded rationality that humans may exhibit and the limited observations they may experience in many settings. To that end, experiments were set up where human subjects would play as followers (adversaries) against each strategy with varying observability conditions under different reward structures. It is not possible to prove optimality against human adversaries who may deviate from the expected optimal responses and thus I rely on empirical validation through experimentation. In addition to examining the performance of previously existing approaches with COBRA(α, ϵ), I also examine the runtime performance of COBRA(α, ϵ) against each of the previous approaches.

4.4.2 Experimental Design

In order to examine these three variables (reward structure, observation condition, algorithm parameters) I constructed a game inspired by the security domain at LAX [Pita et al., 2008] described in Section 2.7, but converted it into a pirate-and-treasure theme. The domain has three pirates – jointly acting as the leader – guarding 8 doors, and an individual human subject would act as a single adversary. The 8 doors model the 8 terminals found at LAX. The interface can be seen in Figure 4.1.

Although COBRA(α, ϵ) allows for multiple follower types (a Bayesian Stackelberg game) each subject was modeled as a single follower type defined by the reward structure they were given. The subject’s goal was to steal a treasure from behind a door without getting caught. Each of the 8 doors would have a different reward and penalty associated with it for both the subjects

as well as the pirates. For instance, as shown in Figure 4.1, door 4 has a reward of 3 and a penalty of -3 for the subject and a reward of 1 and a penalty of -5 for the pirate. If a subject chose a door that a pirate was guarding, the subject would incur the subject penalty for that door and the pirate would receive the pirate reward for that door, else vice-versa. Going back to the previous example, if the subject chose door 4 and a pirate was guarding that door then the subject would receive -3 and the pirate would receive 1. This setup led to a Stackelberg game with $\binom{8}{3} = 56$ leader actions, and 8 follower actions.

4.4.2.1 Participants

There are two different experimental evaluations that were conducted in this study. In the first experimental evaluation there were 178 participants consisting of Engineering undergraduate and graduate students at the University of Southern California. Of the subjects in the first experimental evaluation, 81% were men and ages ranged from 17 to 32 ($M = 22$, $SD = 3$). In the second experimental evaluation there were 40 participants consisting of Engineering undergraduate and graduate students at the University of Southern California. Of the subjects in the second experimental evaluation, 80% were men and ages ranged from 18 to 28 ($M = 22$, $SD = 2$).

4.4.2.2 Reward Structure

One of the variables examined is the reward structure of the pirate-and-treasure domain. Depending on the reward structure human choices could vary vastly. For instance, in some reward structures there may be only a single action that obtains the highest expected value possible, while in a different reward structure there may be multiple different actions that obtain the highest expected value possible. For this study I examine reward structures similar to those used in

the security domain at Los Angeles International Airport (LAX) described previously [Pita et al., 2008].

I constructed four different reward structures for the 8-door 3-pirate domain described. These reward structures can be found in the Appendix under Section B, Tables B.1-B.4. There are three key features in these reward structures. First, the reward scale is similar to that used at LAX to determine the payoffs for both the leader and the follower. Namely, rewards range from 1 to 10 and penalties range from -10 to -1. Second, these reward structures meet the model criteria of what are known as security games as described in Chapter 2 Section 2.6, which are used in domains such as LAX and FAMS Jain et al. [2010]. Finally, in addition to the reward scale and to ensuring that the requirements of a security game were met, I also wanted to examine reward structures where the follower's small ϵ -deviations from the strong Stackelberg equilibrium (SSE) were significantly harmful to the leader (such arbitrarily small ϵ -deviations model cases where the follower does not break ties in favor of the leader).

These reward structures are particularly interesting for the new robust method of COBRA(α, ϵ) since it specifically guards against such potentially harmful deviations. Reward structure four is the baseline case, a zero-sum reward structure, where deviations from the follower's optimal strategy based on a SSE assumption are only better for the leader. In the other three reward structures, the defender's *average* expected value for an ϵ -deviation – for arbitrarily small ϵ – from a SSE is held between -1.30 and -1.96; however, the *worst-case* expected value for a deviation becomes progressively worse for each reward structure. In reward structure three the worst-case deviation is -3.16, in reward structure two it is -4.21, and finally in reward structure one it is -4.56. Thus, for each of these three reward structures, follower's deviations from optimal play based on a SSE assumption can lead to potentially large degradations in the defender's expected value.

As stated before, Figure 4.1 shows the interface used to convey the reward structures to the subjects. The worst-case outcome for deviation from SSE is progressively better from reward structure one to reward structure three. The key question is whether there is any systematicity in human subject's deviations from SSE under different conditions and if $\text{COBRA}(\alpha, \epsilon)$ can capture them sufficiently to mitigate their impact. If so, then it is expected that the robust model will provide the largest benefit in reward structure one and the least benefit in reward structure three. In reward structure four it has been shown that, given no observational uncertainty, $\text{COBRA}(\alpha, \epsilon)$ is equivalent to the optimal maximin strategy and thus there will be no benefit due to this robust approach. However, in all four reward structures it is expected that $\text{COBRA}(\alpha, \epsilon)$'s method for handling observational uncertainty will provide benefits when observational capabilities are low.

4.4.2.3 Observability Conditions

A second variable in this experimental design is different observability conditions for human adversaries. Security officials, such as those at LAX, are interested in the observational capabilities of their adversaries and how this will affect their decisions. In the real-world, some adversaries may be able to take many observations before deciding to act while others may end up having to act with very little information. It is important to understand how this can affect the decisions and actions humans may take.

There are four separate observability conditions that were examined. First I explain what an observation is exactly. Assume that time can be discretized into rounds and on each round the pirates will choose three doors to guard according to their current mixed strategy. For example, examining the UNIFORM strategy, on each round the guards will choose three doors uniformly at random to guard for that round. A single observation then, consists of seeing where the guards

have stationed themselves for a single round and then moving on to the next round where the guards will once again reposition themselves according to their mixed strategy. An example of an observation can be seen in Figure 4.2. On a separate window the observations the subjects had received in the current game were printed for them to refer back to. For instance, after five observations the separate window may list something like [1,2,4][2,4,5][2,6,7][3,7,8][1,3,6] where each triplet is the three doors the pirates were stationed at in each of the five rounds. On each round the subject will either receive an observation or will be asked to choose a door to attack.

The four different observation conditions tested were: (i) The subject does not get any observations (a 1 round game), (ii) the subject gets 5 observations (a 6 round game), (iii) the subject gets 20 observations (a 21 round game), and (iv) the subject gets unlimited observations – simulated by revealing the exact mixed strategy of the pirates to the subject. This final condition is also a 1 round game, but the mixed strategy the pirates used to select their 3 doors is displayed and the subjects are then allowed to choose which door they will attack on this first round. The mixed strategy is revealed as the marginal distribution of guards over the 8 doors. Specifically, for each door the subject is given the probability that any guard will appear on that door for any given round.

As shown in Figures 4.1 and 4.2, subjects were given full knowledge of their rewards and penalties and those of the pirates in all situations. In each game the subject observed the pirates' strategy under the current observability condition, reward structure, and strategy and then was allowed to make his decision on the final round of the game. After making his decision the subject was informed whether he was successful or whether he was caught.

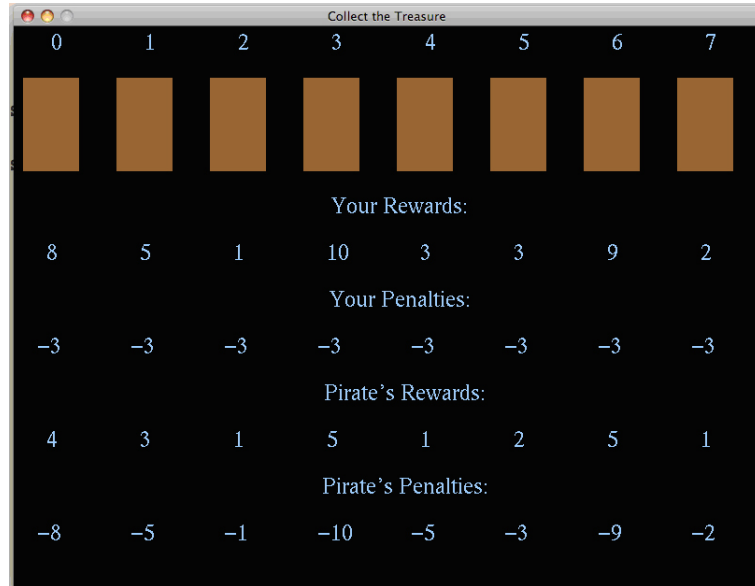


Figure 4.1: Game Interface

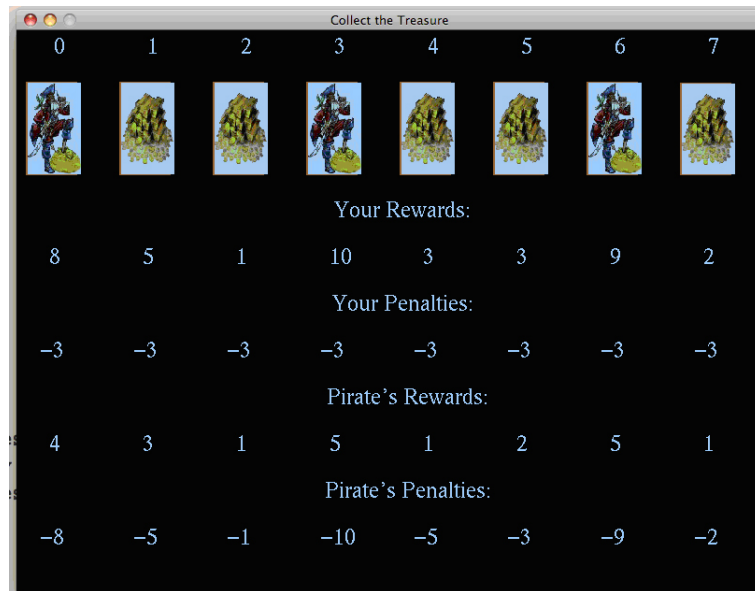


Figure 4.2: Single Observation

4.4.2.4 Algorithms and Parameters

Finally, the last variable examined was the performance of $\text{COBRA}(\alpha, \epsilon)$ against human subjects given adjustments to its α and ϵ parameters under different settings for the first two variables in an attempt to understand how to better deal with human adversaries. These experiments

compare DOBSS, COBRA(0, ϵ), COBRA(α , ϵ), MAXIMIN, and UNIFORM. I chose to include COBRA(0, ϵ) in these experiments to demonstrate the value of having both observational uncertainty and bounded rationality assumptions in an algorithm over having just one. It is important to include either COBRA(α , 0) or COBRA(0, ϵ) to examine whether observational uncertainty, bounded rationality, or the combination of both results has the largest affect on the results. Based on the results in all four conditions for COBRA(0, ϵ) the need to address both bounded rationality and observational uncertainty was found to be important. Furthermore, the second set of experiments were conducted to explore the impact of α .

The value of ϵ in an application should reflect the assumptions regarding the uncertainty in adversary reward and the precision with which the human adversary decides between the different actions. While determining the human adversary's precision is difficult in general, in these experiments the value of ϵ was determined based on three factors: (i) $\epsilon > 0$ based on the assumption that subjects playing these games were not precisely computing expected utilities, (ii) ϵ had to be set to a value to produce qualitatively different leader strategies than created by DOBSS and MAXIMIN to help gain a clear understanding of how ϵ affects the results against human subjects, and (iii) ϵ had to be held constant across the four games. Hence I chose $\epsilon = 2.5$, which lead to 3 to 4 actions in the subject's ϵ -optimal set. This is about the halfway point given these experiments had 8 total choices of actions. DOBSS assumes a single action choice to the adversary and MAXIMIN makes a worst-case assumption. In some settings a higher or lower selection of ϵ may be appropriate. Finding a more precise method for selecting ϵ is left for future studies.

Unlike deciding ϵ , a single choice for α will not hold across all reward structures. First, α could be expected to vary with observability conditions. Second, even for a fixed observability

condition, identical values of α across reward structures is not appropriate. Included in the Appendix under Section F are tables for how $\text{COBRA}(\alpha, 2.5)$ changes as α changes from 0 to 1. Notice that in some reward structures changing α may not necessarily change the mixed strategy for the leader as in Table F.4. For this reason it is necessary to examine each reward structure individually. Within each reward structure, I tried two different techniques to choose α , one with a fixed α and one with a variable α , in order to compare the impact of variable α more clearly.

Since the value of α is used to balance the amount of observed information and a priori bias information that the adversary incorporates in his assumption of the leader strategy, this parameter should be related to the amount of observations made by the follower. Thus, the technique with variable α is obviously the more standard version. Clearly when the follower has unlimited information $\alpha = 0$ (the follower correctly estimates the leader strategy) and when he has no observations $\alpha = 1$ (the follower uses the a priori bias). Less straight forward is how to set an α value when the follower has 5 or 20 observations. I follow two methods of adjusting α :

- The first is to conduct r trial experiments with human adversaries using $\text{COBRA}(0, \epsilon)$ with 0, unlimited, and n observations. In these experiments n is either 5 or 20. Given the choices made by each subject in the trial experiments, r subject expected values are collected for each observation condition. Let $\text{corr}_{n,0}$ be the correlation between the r subject expected values for the n observation condition and the 0 observation condition. Similarly let $\text{corr}_{n,u}$ be the same with the unlimited observation condition. α is set for n observations as $\alpha = \text{corr}_{n,0} / (\text{corr}_{n,0} + \text{corr}_{n,u})$. If the results were more correlated with the unobserved condition this would make α higher and otherwise it would make α lower. It is assumed

that having more observations would lead to results that were more correlated with the unlimited observation condition and having less observations would lead to results that were more correlated with the unobserved condition. I use this form of adjusting α in reward structures 1 and 2.

- The second method to set α uses fixed arbitrary values: $\alpha = .75$ for the 5 observation condition and $\alpha = .25$ for the 20 observation condition. This simple method was created to evaluate the necessity of conducting trial runs to determine the α values experimentally. I use this second method in reward structures 3 and 4.

Finding an exact method to select α in these conditions remains an issue for future studies. Before running this study there was no evidence to suggest either method attempted would work best. However, I hoped to gain some insight into choosing α based on the results obtained from both approaches. Section 4.4.4 provides more analysis for choosing α .

The second technique to selecting α is to assume a constant α , leading to a version of COBRA(α, ϵ) that I will refer to as COBRA(C, ϵ) (COBRA(α, ϵ) with constant α). For COBRA(C, ϵ), α was set to the same α value as the 5 observation condition from the four reward structures with the expectation that it would perform poorly in higher observation conditions since it was not appropriately adjusted. The α values chosen for COBRA(α, ϵ) in each of the reward structures are summarized in Table 4.1. In Section 4.4.4 I will present experimental results for many additional settings of the α parameter in both the unobserved and unlimited observation conditions. Based on these settings I will introduce a third, better performing method for determining α that generalizes to all four reward structures and is more practical in the field.

	Unobserved	5	20	Unlimited
Structure One	1	.37	.03	0
Structure Two	1	.54	.41	0
Structure Three	1	.75	.25	0
Structure Four	1	.75	.25	0

Table 4.1: α values used in each observation condition for a given reward structure

4.4.2.5 Experimental Procedure

For the first experiment, each of the 96 game settings (four reward structures, six algorithms, and four observability conditions) were played by 40 unique subjects (i.e., in total there were 2480 total trials). There are fewer than 3840 trials however because some are duplicates. For example, the unobserved condition for a particular reward structure need only be played by 40 subjects as opposed to 240 subjects (one set of 40 for each algorithm). The choices made by those 40 subjects could then be used to compare all six algorithms since the subjects were not given any information about a particular algorithm in advance for this condition. Thus the choices made were irrelevant of the algorithm used. Also notice that in some observation conditions a few of the algorithms yielded the same strategy (e.g., $\text{COBRA}(\alpha, \varepsilon)$ and $\text{COBRA}(C, \varepsilon)$ in the five observation condition used the same ε and α parameters).

The 96 potential game settings were broken into two separate groups of 48 game settings. The first group of 48 game settings consisted of all combinations of game settings for reward structures 1 and 2 (i.e., 2 reward structures, six algorithms, and four observability conditions). Similarly, the second group consisted of the same for reward structures 3 and 4. All experiments were first conducted for group 1 and then group 2. The following procedure was used for both groups.

Given these 48 game settings, each subject played a total of 14 unique games. For now I will omit the 2 game settings where the subjects play the unobserved observation condition for each

reward structure. This leaves 46 potential game settings. For every 10 subjects, 14 games were chosen completely at random without replacement from the 46 possible remaining game settings. This single random ordering was played by all 10 subjects. Once a particular game setting was played by 40 subjects it was removed completely from the set of possible game settings. If less than 14 potential game settings remained, the remaining games were chosen at random from the completed game settings, but not recorded. For example, if there were only 12 potential game settings that did not have 40 total subjects, then 2 game settings would be chosen from the other 44 game settings that did have 40 total subjects, but the results for these 2 games would not be recorded.

The first 40 subjects played the 2 game settings that I previously omitted. In this case, the first two games played were always these 2 game settings (i.e., the unobserved observation condition for each reward structure). This was done so they would not have viewed any strategies that could influence their anchoring bias. After these 2 games, the next 12 were chosen for every 10 subjects using the same procedure outlined above. I presented the games in random orderings in an attempt to reduce ordering effects. Subjects were not allowed to participate in this study more than once.

The actual experiments were conducted in a campus office on two standard desktops. When subjects came in they were presented with the instructions seen in the Appendix under Section G.1. Subjects were given an unlimited amount of time to study the instructions and ask questions before beginning to ensure they understood how the game was played. Once subjects began they were also given an unlimited amount of time to complete the game.

For each game, the objective of a subject was to earn as many points as possible. The subject was allowed to choose a single door, based on the current reward structure and observation condition, that they believed was unguarded and once a door was chosen that game was over and the subject played the next game. As stated previously, at the end of each individual game the subject was informed whether he was successful or not. Starting with a base of US \$8.00, each reward point within the game was worth US \$0.15 for the subject and each penalty point deducted US \$0.15. This was incorporated to give the subjects incentive to play as optimally as possible. For a given algorithm the leader expected value was computed for each follower action (i.e., for each choice of door by subject). Then, the average expected value was calculated for a given algorithm using the actual door selections from the 40 subject trials.

For the second experiment, there were only 12 game settings in total. While I will describe in detail these 12 game settings in Section 4.4.4, for these experiments I used the exact same setup described above, but the subjects only played 12 games instead of 14. Again, each set of 10 subjects was presented one random ordering of the 12 games.

4.4.3 Experimental Results

Figure 4.3(a) shows the average leader expected value for the first reward structure, with each data-point averaged over 40 human responses. Figures 4.3(b), 4.3(c), and 4.3(d) show the same

for the second, third, and fourth reward structures¹. In both figures, the x -axis shows the observation condition for each strategy and the y -axis shows the average expected value each strategy obtained. For example, examining Figure 4.3(a) in the unlimited observation condition, COBRA(C, ϵ) scores an average leader expected value of -0.33, whereas DOBSS suffers a 663% degradation of expected value, obtaining an average score of -2.19.

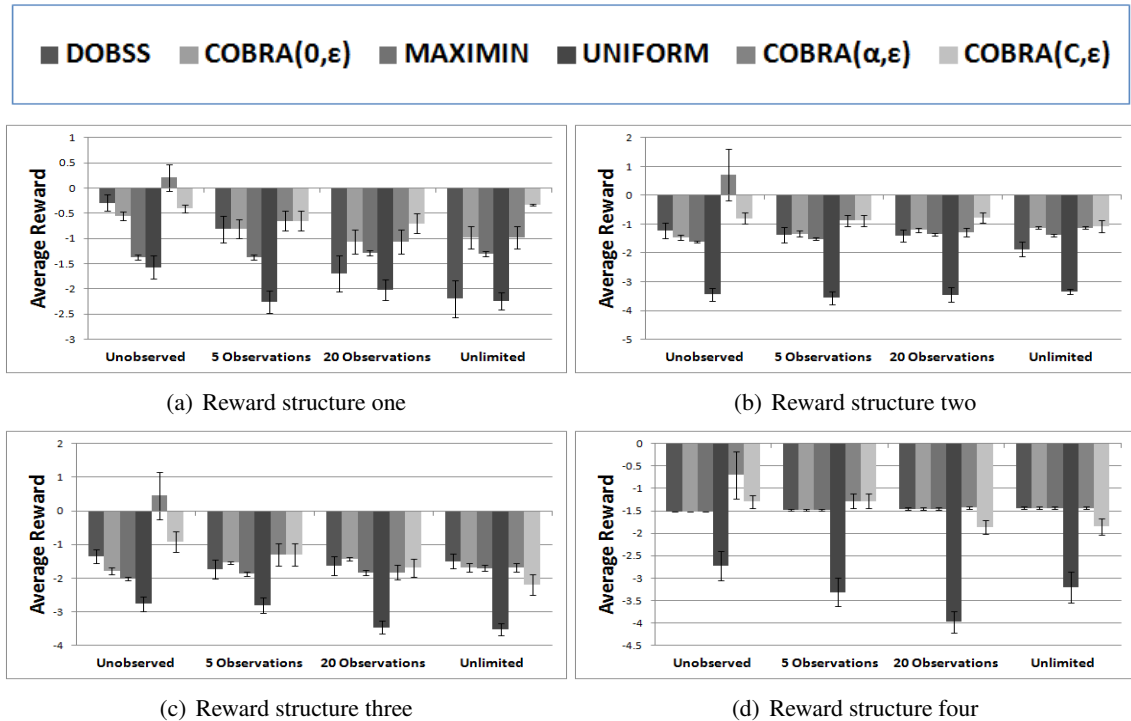


Figure 4.3: Average leader expected value

¹The reason all strategies obtain a negative average is due to the lack of enough resources (or guards) in this setting, but that is where randomization strategies have the most impact. The reason COBRA(α, ϵ) was able to obtain a positive average expected value in the unobserved condition is that it placed all its resources on a small subset of doors (a more deterministic strategy) assuming that humans would choose these doors based on their belief of the ignorance prior (uniform distribution). In practice the humans did indeed play according to this expectation and thus the expected values obtained were much higher. Examining Tables D.1-D.4 in the Appendix under Section D, it is clear that while most expected values are negative, in the unobserved condition for COBRA(α, ϵ) (seen as COBRA(1,2.5)) there are a few distinct doors that obtain very high expected values, which correspond to the doors chosen by most human subjects in these experiments for this condition.

4.4.3.1 Key Observations

I provide my key observations first, then provide statistical significance tests in the following section and later provide a deeper analysis. The *main* observation from Figure 4.3 is that the $\text{COBRA}(\alpha, \varepsilon)$ algorithm has a performance that is superior to the theoretically optimal DOBSS algorithm and baseline approaches against human subjects. I can breakdown this main observation into observations of the following trends:

1. *Considering observational uncertainty is important:* When creating algorithms for leader strategies in Stackelberg games it is important to address biases that may arise from observational uncertainty. Focusing first on the unobserved and 5 observation conditions, $\text{COBRA}(\alpha, \varepsilon)$ and $\text{COBRA}(C, \varepsilon)$ obtain much better results demonstrating the benefit of incorporating an anchoring bias. In the 20 and unlimited observation conditions it can be seen that $\text{COBRA}(\alpha, \varepsilon)$ and $\text{COBRA}(C, \varepsilon)$ can still provide benefits, however, in these cases it is assumed that observational uncertainty is low so the effect of anchoring biases begin to diminish. As can be seen in Figure 4.3, $\text{COBRA}(\alpha, \varepsilon)$ performs better than DOBSS in all except only two cases (Reward structure three under 20 and unlimited observations). Because of its lack of adjustment of α , $\text{COBRA}(C, \varepsilon)$ is seen to perform better than DOBSS in all conditions except in four total cases confined to reward structures three and four.
2. *Addressing bounded rationality is an important component when designing algorithms that perform against humans:* Examining $\text{COBRA}(0, \varepsilon)$ specifically, it can be seen in the 20 and unlimited observation conditions for reward structure one and two that addressing bounded rationality provides improvements over DOBSS. In the lower observation conditions we

begin to see how observational uncertainty becomes a larger factor making bounded rationality a secondary issue. This follows from the first observation presented. In reward structure four, a zero-sum game, $\text{COBRA}(0,\varepsilon)$ and DOBSS are exactly equivalent, so no gains are expected. In reward structure three, where deviations from SSE were least harmful of the remaining reward structures, there is some difference between $\text{COBRA}(0,\varepsilon)$ and DOBSS, but not as much as in reward structures one and two. In general though, these results confirm the expected result that humans do not strictly play the game theoretic optimal strategy. Thus, it is important to address deviations from this theoretic optimal strategy based on bounded rationality to prevent what could potentially be significant losses.

3. *COBRA(C,ε) surprisingly outperforms COBRA(α,ε) under high observation conditions in some reward structures:* This unexpected result led to the subsequent experiments reported in Section 4.4.4 and to a more efficient heuristic for selecting α .

4.4.3.2 Statistical Significance

The main observation in the previous section critically depends on significant differences between $\text{COBRA}(\alpha,\varepsilon)$ and the remaining strategies. I chose to employ more robust statistical methods for these tests in order to overcome limitations with the data set. These limitations include a non-normal distribution (due to a very small number of discrete choices as opposed to continuous or near continuous choices) and high variance. Having a normal distribution is an important assumption of traditional statistical tests such as the classic T-test.

For the statistical significance tests I used a one-way Brunner-Puri test [Brunner et al., 1999] for repeated observations in the unobserved condition and I used Yuen's test for comparing

trimmed means [Yuen, 1974] in the 5, 20, and unlimited observation conditions. In the unobserved condition all structures were treated separately, however, in the 5, 20, and unlimited observation conditions reward structures one and two were combined into a single data set. For an in depth discussion of this decision and also why these statistical tests were chosen please see the Appendix, Section A. In general, given the nature of the data — discrete rather than continuous distribution of values and non-normal distributions — it can be difficult to obtain statistical significance without significantly larger data sets. Yet, these results do achieve statistical significance in key cases *demonstrating the effectiveness of the COBRA(α, ϵ) strategy*, as summarized below.

Conclusions regarding unobserved condition: Looking first at the unobserved condition, COBRA(α, ϵ) obtained statistical significance against DOBSS in reward structures one, two, and three with a maximum p-value of .04. Since reward structure four is a zero-sum game I reiterate that the strategy space for COBRA(α, ϵ) is limited. Namely, in all observation conditions the results for the DOBSS, MAXIMIN, and COBRA(0, ϵ) algorithms are identical. Thus, the only way to alter the strategy based on this robust method is through the use of α . Although the results are in favor of COBRA(α, ϵ) in this reward structure, in the unobserved condition it was not possible to achieve statistical significance against DOBSS/COBRA(0, ϵ)/MAXIMIN without a larger data set. Against MAXIMIN and UNIFORM, COBRA(α, ϵ) obtains statistical significance in reward structures one, two, and three with a maximum p-value of .04 except in reward structure one where it obtains a p-value of .098 against MAXIMIN. Overall these results demonstrate the superiority of COBRA(α, ϵ) over DOBSS and simple baseline algorithms in the unobserved condition.

Conclusions regarding combined data from reward structures one and two in remaining observation conditions: In the 5, 20, and unlimited observation cases the maximum p-value obtained for $\text{COBRA}(C, \epsilon)$ versus any other strategy was .033. Given that $\text{COBRA}(C, \epsilon)$ is shown outperforming every other strategy, including $\text{COBRA}(\alpha, \epsilon)$, under these observation conditions, this establishes that $\text{COBRA}(C, \epsilon)$ is statistically significantly better than all other strategies in these reward structures under these observation conditions. This in turn demonstrates the superiority of the $\text{COBRA}(\alpha, \epsilon)$ algorithm as a whole in these reward structures and observation conditions since $\text{COBRA}(C, \epsilon)$ is an instantiation of $\text{COBRA}(\alpha, \epsilon)$ with a particular choice of α .

Conclusions regarding reward structure three in remaining observation conditions: In reward structure three for the 5 and 20 observation conditions we do not achieve statistical significance between DOBSS, $\text{COBRA}(\alpha, \epsilon)$, and $\text{COBRA}(C, \epsilon)$ making the results obtained inconclusive. To a certain extent this is an implication of deviations in reward structure three not being as harmful to the leader (see Section 4.4.2.2) and thus more difficult to obtain significant differences between the strategies. However, in the unlimited observation condition we find that DOBSS is statistically significantly better than all other strategies with a maximum p-value of .036. Although in the unlimited condition, given the choices made for α and ϵ , DOBSS outperforms the robust strategy of $\text{COBRA}(\alpha, \epsilon)$, I will later present an alternative choice for these parameters in Section 4.4.4 that is able to outperform DOBSS.

Conclusions regarding reward structure four in remaining observation conditions: In reward structure four, as presented in Section 4.3, it has been shown that MAXIMIN, DOBSS, and $\text{COBRA}(0, \epsilon)$ are equivalent. In the unlimited and 20 observation conditions, given the choice for α , it is also the case that $\text{COBRA}(\alpha, \epsilon)$ and DOBSS/MAXIMIN/ $\text{COBRA}(0, \epsilon)$ are equivalent. Given these equivalences, our only concern is $\text{COBRA}(C, \epsilon)$. Since $\text{COBRA}(C, \epsilon)$ is outperformed

in the 20 and unlimited observation conditions it provides no benefits in these cases. However, for the 5 observation condition COBRA(C, ϵ) (or COBRA(α, ϵ)) is statistically significantly better than all other strategies with a maximum p-value of .005. Given these results and the statistical significance achieved in the 5 observation condition, COBRA(α, ϵ) is found to be the superior strategy, even in a zero-sum game, due to its ability to handle observational uncertainty.

4.4.3.3 Analysis of Results

I discuss the key implications of the observations presented in Section 4.4.3.1 and why they were reached. I include two tables for reference in the following discussion, Tables 4.2 and 4.3. Table 4.2 shows the expected values (for a subset of the algorithms tested) the leader should obtain for each door selection by the follower in reward structure one on average. For instance, if the follower selected Door 2 when playing against DOBSS the leader would expect to obtain an expected value of -.97. Obviously depending on whether there was a guard stationed there or not in any particular instance the leader would get the respective reward or penalty associated with that door, but over time the average would converge to the expectation of -.97.

I have placed in bold font the predicted expected value for each of the algorithms. This predicted expected value is the expected value an algorithm expects to receive based on the assumptions it has made. For example, DOBSS is an algorithm that expects the follower to play a strong Stackelberg Equilibrium (SSE) strategy. This means that the follower will choose, between his highest expected value choices, the door that is also best for the leader. This SSE strategy is the expected value that has been placed in bold font for DOBSS. MAXIMIN on the other hand is an algorithm that makes a worst-case assumption and thus all doors that give the minimum expected value are placed in bold font.

Table 4.3 shows the percentage of times the follower chose a response that gives the leader (pirate) an expected value equivalent to or higher than the predicted expected value for the current algorithm under different observation conditions in reward structure one. I will refer to these responses as *expected strategy(s)*. To clarify what I mean by *expected strategy(s)* I will look at COBRA(0, ϵ) as an example. As seen in Table 4.2, this algorithm expects to receive an expected value of -.36. Thus, in Table 4.3 under the unobserved observation condition it can be seen that the follower chose a door that gave the leader an expected value of -.36 or higher (an *expected strategy*) 65% of the time. I point out that MAXIMIN is a strategy that expects a worst-case outcome and thus all doors are *expected strategies*. DOBSS is on the opposite extreme, since it assumes perfectly rational play, and thus generally results in a single door being the *expected strategy*. COBRA(0, ϵ) and COBRA(α , ϵ) fall somewhere in between these lines, where multiple doors within the ϵ -optimal strategies are *expected strategies*, but less doors than MAXIMIN where all doors are *expected strategies*. Of course as shown in Section 4.3 this depends on the setting of ϵ since a setting that is too high will yield the same result as MAXIMIN.

Table 4.2 shows an important trade-off. MAXIMIN achieved a 100% match with *expected strategies* (Table 4.3), but it does so by making all leader expected values low (-1.63). DOBSS achieves low match with *expected strategies*, but its leader predicted expected value is higher (.39). COBRA(α , ϵ) is in the middle of these extremes. Included in the Appendix under Sections C, D, and E respectively are: i) tables presenting the actual mixed strategies for each of the reward structures, ii) tables of the expected values for each reward structure given the strategies presented in i), and iii) tables for the percentage of times the follower chose an *expected strategy* in each reward structure. In each of the expected value tables I have placed in bold font the predicted expected value for each of the algorithms. I reiterate that an *expected strategy* is any

door selection that gives an expected value for the leader at least as high as the values in bold font.

	DOBSS	COBRA(0, ϵ)	MAXIMIN	COBRA(.37, ϵ) COBRA-5	COBRA(.03,2.5)
Door 1	-5	-4.58	-1.63	-5	-4.61
Door 2	-.97	-.42	-1.63	-.30	-.37
Door 3	.36	-.36	-1	-.30	-.37
Door 4	-1.38	-.79	-1.63	-.30	-.73
Door 5	.06	-.36	-1.63	-.30	-.37
Door 6	-1	-.86	-1	-1	-.87
Door 7	.39	-.36	-1.63	-.30	-.37
Door 8	-4.57	-3.69	-1.63	-3.32	-3.67

Table 4.2: Leader expected values for each door selection in reward structure one

Structure One	Unobserved	5	20	Unlimited
DOBSS	20%	7.5%	17.5%	12.5%
COBRA(0, ϵ)	65%	65%	65%	70%
COBRA(α , ϵ)	57.5%	92.5%	72.5%	70%
COBRA(C, ϵ)	92.5%	92.5%	87.5%	95%
MAXIMIN	100%	100%	100%	100%

Table 4.3: Percentage of times follower chose an *expected strategy* in reward structure one

My first conclusion was that observational uncertainty is important. By accounting for this uncertainty, strategies are able to exploit human perceptions and make more appropriate use of resources. That is why COBRA(α , ϵ) performs better than DOBSS in most conditions. In fact, examining Table 4.3 it can be seen that in the unobserved condition against COBRA(α , ϵ), human subjects played an *expected strategy* 57.5% of the time, such as door 3 or door 5 as seen in Table 4.2, while against DOBSS they played an *expected strategy* merely 20% of the time. However, based on MAXIMIN's performance it is clear that getting followers to play *expected strategies* is not the only component.

As mentioned earlier, there is a trade-off in this match with *expected strategies* and leader expected values. MAXIMIN is too loose with its resources making all responses *expected strategies* and thus the benefits begin to diminish because resources are spread too thin. It is important to utilize resources efficiently and not squander them unnecessarily. By more accurately modeling human responses the new strategies of $\text{COBRA}(\alpha, \epsilon)$ are better able to utilize resources to guard against these responses and thus achieve a higher average expected value, one that is closer to the expected value they expect to receive based on their *expected strategies*. Of course when observation is high it is assumed that observational uncertainty is low and the strategies are adjusted accordingly. Thus I find that utilizing a strategy that exploits human anchoring bias, but does not squander resources, provides the benefits I am seeking.

My second conclusion was that addressing bounded rationality is important when dealing with human adversaries. I reach this conclusion due to $\text{COBRA}(0, \epsilon)$'s superior performance. In fact, examining Table 4.3 it is clear that under all observation conditions the assumptions made by DOBSS are a poor model of human choices. Against $\text{COBRA}(0, \epsilon)$ on the other hand, which addresses bounded rationality by utilizing the concept of ϵ -optimal responses, human subjects consistently play *expected strategies* 65-70% of the time under all observation conditions. While this improvement in *expected strategy* match comes at the cost of lower expected value (Table 4.2) the overall results indicate that the trade-off leads to an overall better performance of $\text{COBRA}(0, \epsilon)$.

This is a clear indication of the benefits that can be obtained by addressing bounded rationality. In fact, it can specifically be seen in the 20 and unlimited observation conditions for reward structure one and two that addressing bounded rationality provides improvements over DOBSS. Indeed, it is necessary to address bounded rationality when dealing with humans [Simon, 1956].

Many times their choices can be guided by their cognitive limitations and thus it is necessary to robustly guard against a spectrum of possible choices rather than optimize against a single optimal choice [Simon, 1956; Rubinstein, 1998; Simon, 1969; Camerer, 2003]. By optimizing against the perfectly rational choice DOBSS may make poor use of its resources when dealing with human adversaries. Even COBRA(0, ϵ) is not a perfect model of human behavior, however, it is at least a step in the right direction since it is able to obtain *expected strategies* more often than algorithms without bounded rationality.

I defer the explanation of the final key observation to Section 4.4.4. However, I point out that due to the poor performance of COBRA(α,ϵ) in reward structure three compared to DOBSS I ran further experiments exploring different α values that are separate from the experiments I will present in Section 4.4.4. In these additional experiments, using a strategy with $\alpha = .5$ I found through experimentation with 40 new subjects that in the unlimited observation condition, COBRA(α,ϵ) obtains an average expected value of -1.35 outperforming DOBSS with an average expected value of -1.5. In Section 4.4.4, in addition to $\alpha = .5$ for reward structure three, I present 3 alternate choices for α in each reward structure for the unlimited observation condition on top of the three choices presented in these original experiments.

4.4.4 Handling Observational Uncertainty

Given the significant impact of α on these results, this section provides further analysis of the choice of α on performance. Given that human choice under uncertainty remains a key area of research in psychology [Fox and Clemen, 2005; Fox and Rottenstreich, 2003; Kahneman and Tversky, 1972; Koehler and James, 2009; Starmer, 2000; Tversky and Koehler, 1994], it is difficult to provide a definitive answer; however, I provide a solid initial grounding and heuristics for

choosing α . I focus on the two extreme observability cases — the unobserved observation case and the unlimited observation case — for this initial investigation.

As explained before, since the choices made in the unobserved condition were made irrespective of the strategy used (subjects did not have any information on the strategy being employed when they made their decision) it is possible to test all α values in this case using the data collected for each reward structure. The results are presented in Figure 4.4 and demonstrate a clear increasing trend in all four reward structures. On the x-axis I vary the value of α and on the y-axis I show the average expected value obtained for a particular value of α given the choices made by the subjects in the unobserved condition. For example, looking at Figure 4.4(b), when $\alpha = 0$ the average expected value is -1.46 while when $\alpha = 1$ the average expected value is .7. These results suggest that in the absence of observations, humans do appear to be anchoring on the uniform distribution and thus $\alpha = 1$ is the optimal setting in all four reward structures. It follows that if the expectation is for humans not to take any observations of the leader strategy, exploiting their anchoring biases can be important.

Determining an appropriate α value for the unlimited observation condition is more difficult. Since the humans are given the strategy in advance under this condition and normally (with a few exceptions) changing α also alters the strategy used, it is required to test each new value of α with a new set of subjects. To avoid exhaustively testing a large number of α values, I focused on testing three new α values per reward structure in particular. Again, for these experiments I used the exact same setup described previously but the subjects only played 12 games instead of 14 (for the 12 new α values) and each of these games were with unlimited observation.

The new values of α tested against human subjects were selected using two key criteria. Before discussing these criteria, I present Figure 4.5, which helps ground these criteria. I also

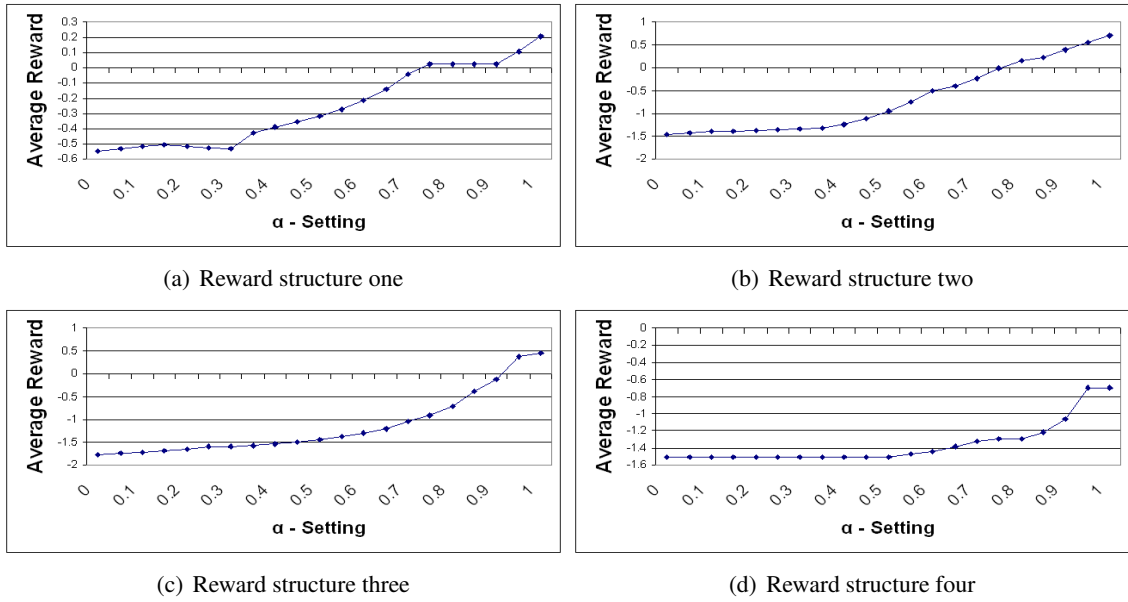


Figure 4.4: Unobserved condition - Expected average reward

define a term which I will call *strategy entropy* as $-\sum_{i=1}^n (p_i * \log(p_i))$ where p_i is the probability value shown on door i . Although this is the standard equation for entropy it differs in this setting as it is defined over the marginal distribution subjects are shown. This is in contrast to calculating the entropy of the mixed strategy that produced this marginal distribution. As explained previously in Section 4.4.2.3, in the unlimited observation condition, the strategies are presented to subjects as the marginal probability distribution of guards over the 8 doors, where the sum of the probabilities over all 8 doors will be 3. Specifically, for each door the subject is given a probability p and this is the probability that he will obtain his penalty (there will be a guard on the door) where $1 - p$ is the probability he will obtain his reward (there will not be a guard on the door).

Based on this definition, a higher *strategy entropy* represents a strategy where the probability value of each door is closer to .375 (300% divided evenly among 8 doors) and a lower *strategy entropy* represents a strategy where the probability value on each door is closer to 1 or 0 (more specifically the highest entropy possible is 4.245). Figure 4.5 shows, for each value of α in each

reward structure, the *strategy entropy* produced from the corresponding mixed strategy. On the x-axis I list the value of α and on the y-axis I list the *strategy entropy* obtained from the corresponding strategy. I highlight the values used in these additional experiments and the original experiments for the unlimited observation condition with a bold circle. More specifically, for each of these reward structures there are 5 highlighted values corresponding to the three new values selected for these additional experiments, the value used in COBRA(C, ϵ) in the original experiments, and finally the value used in COBRA(α , ϵ) in the original experiments which was $\alpha = 0^2$.

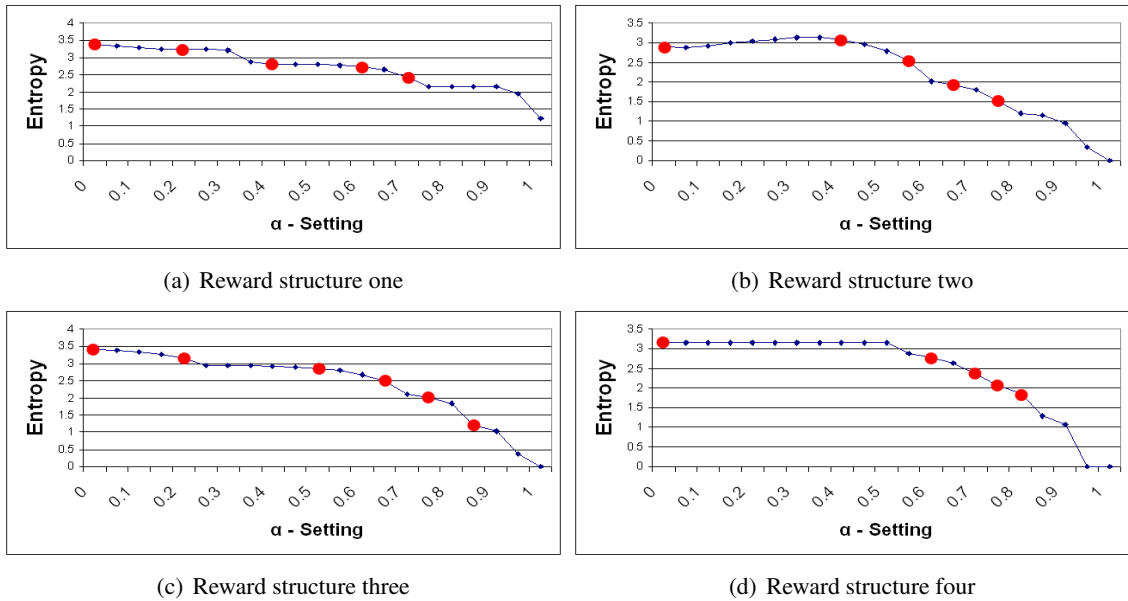


Figure 4.5: *Strategy entropy* for varying α values

Given Figure 4.5, the two criteria for selecting α values are as follows: i) The *strategy entropy* for the corresponding mixed strategy should not be low as this corresponds to deterministic strategies. Since humans are provided this strategy in advance, they would certainly exploit a

²In reward structure three there is a sixth value that was used in the original experiments as an alternative value for COBRA(C, ϵ) in the unlimited observation condition to outperform DOBSS.

very high α , i.e., low entropy. I selected strategies with entropy 1.19 or higher; and ii) The *strategy entropy* for the corresponding mixed strategy should be quantitatively different than other α values already being tested. For example, for reward structure four, strategy entropy is constant over a range of α values as the strategy is constant. Selecting two α values from this range is not useful.

The results of these new experiments along with the results from the original experiments in the unlimited observation condition can be seen in Figure 4.6. On the x-axis I present the different algorithms and different values of α for COBRA(α, ϵ) and on the y-axis I present the average expected value obtained by each of the corresponding strategies. For example in Figure 4.6(b), COBRA(.54,2.5) obtains an average expected value of -1.08.

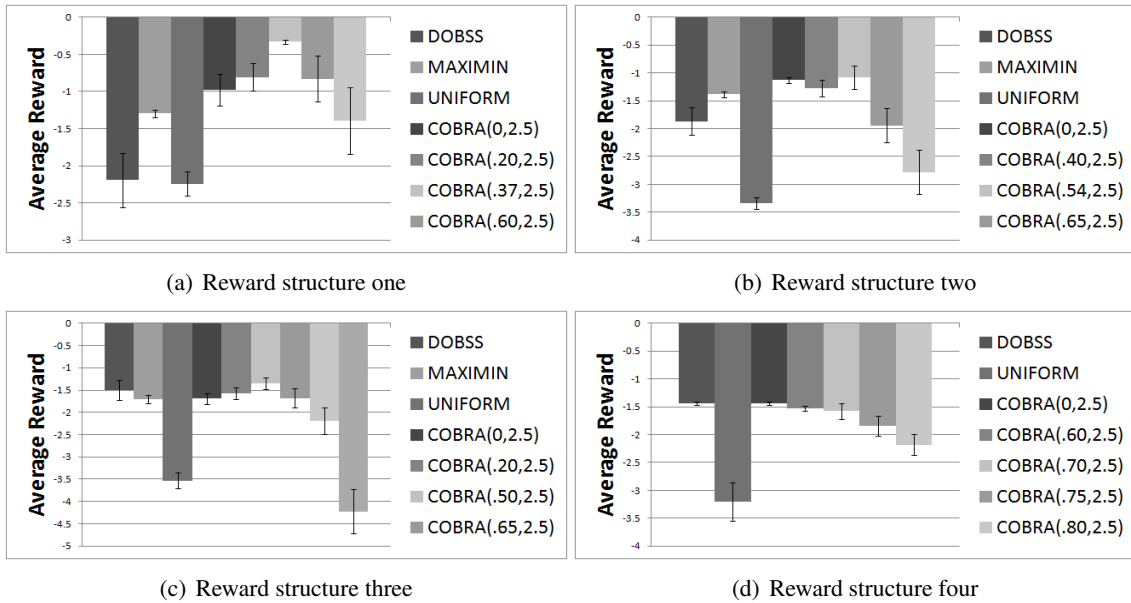


Figure 4.6: Average expected values for varying α under the unlimited observation condition

Figure 4.6 allows the following observations about setting α in COBRA (α, ϵ) for the unlimited observation condition to be made:

- *High values of α lead to poor performance:* Choosing a high α , which leads to a lower strategy entropy performs poorly. Lower strategy entropy implies more determinism, which human followers exploit. In these experiments, strategy entropy below 2.4 appeared to degrade performance.
- *Mid-range values of α may lead to the best performance in reward structures where a follower's deviation from expectation is harmful to the leader:* Previously it was assumed that $\alpha = 0$ would be the best setting for the unlimited observation condition, but these results are clearly contradictory to this. Choosing the lowest values of α (near 0) leads to a high strategy entropy, but that does not provide the best outcome (in three of the four reward structures). One possible explanation for this result is that humans have difficulty reasoning about strategies that are overly complicated. If strategy entropy is too high then humans may have difficulty evaluating alternatives in order to find optimal or even ϵ -optimal strategies and this may cause them to deviate from *expected strategies* which can lead to a degradation in the leader's expected value — as explained previously, these three reward structures were specifically designed such that deviation by the followers from *expected strategies* were harmful to the leader. In these experiments, a strategy entropy between 2.5 and 2.85 appears to be optimal.
- *Lowest values of α may perform optimally in zero-sum reward structures:* In the fourth reward structure, or zero-sum reward structure, the α value that gives the maximum strategy entropy (i.e., COBRA(0,2.5)) is optimal. This makes intuitive sense given that the optimal strategy in a zero-sum game is equivalent to a MAXIMIN strategy as shown by the observations in Section 4.3. In a zero-sum game, any deviation from optimal play by the follower

leads to an expected value that is strictly higher for the leader. For reward structures one, two, and three however, deviations can lead to severe degradations in the leader's expected value, making it more important to account for deviations from optimal play. The robust strategies in $\text{COBRA}(\alpha, \epsilon)$ are designed to combat these potentially detrimental deviations, which means in a zero-sum game, as previously explained, they are the least useful. However, even in a zero-sum game it is still possible to exploit human anchoring biases in low observation conditions to obtain higher expected leader values as shown by these results.

Based on these results we can see why $\text{COBRA}(C, \epsilon)$ is seen to outperform $\text{COBRA}(\alpha, \epsilon)$ in the unlimited observation condition as noted in Section 4.4.3.1. Also, these results have demonstrated that the optimal choice for α in the unobserved observation condition appears to be $\alpha = 1$. In the unlimited observation condition these results have shown that, for general sum games, setting α to a value that leads to a strategy between a mid-range to high-range entropy appears to be best. While a precise predictor for optimal α remains an issue, particularly for other observation conditions, using a *strategy entropy* based technique for selecting α in these conditions appears to be a promising approach.

Given the analysis presented in Section 4.4.3.3 and these additional experiments, $\text{COBRA}(\alpha, \epsilon)$ and $\text{COBRA}(C, \epsilon)$, with appropriately chosen α and ϵ values, appear to be the best performing among the presented algorithms. The performance of DOBSS in some of these experiments also illustrates the need for the novel approaches presented in this thesis for dealing with humans. For example, in Figure 4.3(a) it is clear that under high observation conditions DOBSS performs very poorly in comparison to other strategies. In fact, in this case DOBSS is seen performing even worse than simple baseline algorithms such as MAXIMIN. Indeed, with DOBSS having been deployed since August 2007 at Los Angeles International Airport (LAX) [Jain et al., 2010],

these results show that security at LAX could potentially be improved by incorporating these new methods for dealing with human adversaries.

4.4.5 Runtime Results

For runtime results, in addition to the original 8-door game, I constructed a 10-door game with $\binom{10}{3} = 120$ leader actions, and 10 follower actions. To average the run-times over multiple instances, I created 19 additional reward structures for each of the 8-door and 10-door games. Furthermore, since the algorithms presented handle Bayesian games, I created 8 variations of each of the resulting 20 games to test scale-up in number of follower types. For the *a priori* probability distribution of follower types, I assume each follower type occurs with a 10% probability except the last which occurs with $1 - .10(n - 1)$ probability where n is the number of follower types. For example, if there are 5 follower types, the first four types each occur with a 10% probability and the last type occurs with a 60% probability. Experiments were run using CPLEX 8.1 on an Intel(R) Xeon(TM) CPU 3.20GHz processor with 2 GB RDRAM.

In Figure 4.7, I summarize the runtime results for the Bayesian game using DOBSS, COBRA(0, ϵ), COBRA(α ,0), COBRA(α , ϵ) and MAXIMIN. I include one graph for the 8-door results and one for the 10 door results. For COBRA(α , ϵ) I set $\epsilon = 2.5$. For both COBRA(α ,0) and COBRA(α , ϵ) I varied the value of α to show the impact on solution speed. I include $\alpha = .25$ and $\alpha = .75$ in the graph, denoted by COBRA(.25,2.5)/COBRA(.75,2.5) for COBRA(α , ϵ) and COBRA(.25,0)/COBRA(.75,0) for COBRA(α ,0) respectively. The x -axis in Figure 4.7 varies the number of follower types from 1 to 8. The y -axis of the graph shows the runtime of each algorithm in seconds. All experiments that were not concluded in 20 minutes (1200 seconds) were cut off.

As expected, MAXIMIN is the fastest among the algorithms with a maximum runtime of 0.054 seconds on average in the 10-door case. Not anticipated was the approximately equivalent runtime of DOBSS and COBRA(0, ϵ) and even more surprising were the significant speedups of COBRA(α,ϵ) and COBRA($\alpha,0$) over DOBSS and COBRA(0, ϵ) depending on the value of α . As shown in Figure 4.7 as α increases, the runtime of COBRA(α,ϵ) and COBRA($\alpha,0$) decreases. For example, in the 10-door 8 follower type case when $\alpha = .25$ COBRA(α,ϵ) is unable to reach a solution within 1200 seconds on average, however, when α is increased to .75, COBRA(α,ϵ) is able to find a solution in 327.5 seconds on average. In fact, excluding MAXIMIN, every strategy except COBRA(α,ϵ) with $\alpha = .75$ and COBRA($\alpha,0$) reached the maximum runtime in the 10-door 8 follower type domain. This speedup could be attributed to the branch-and-bound methods used to find solutions to these MILPs. Since COBRA(α,ϵ) and COBRA($\alpha,0$) distribute some of their weight to the uniform distribution it decreases the number of branch-and-bound nodes necessary to achieve a solution by decreasing the branch space. This set of results particularly support this theory. These results demonstrate that COBRA(α, ϵ) does not incur significant runtime costs for the proposed enhancements to deal with bounded rationality and observational uncertainty and in fact may even provide runtime improvements over DOBSS.

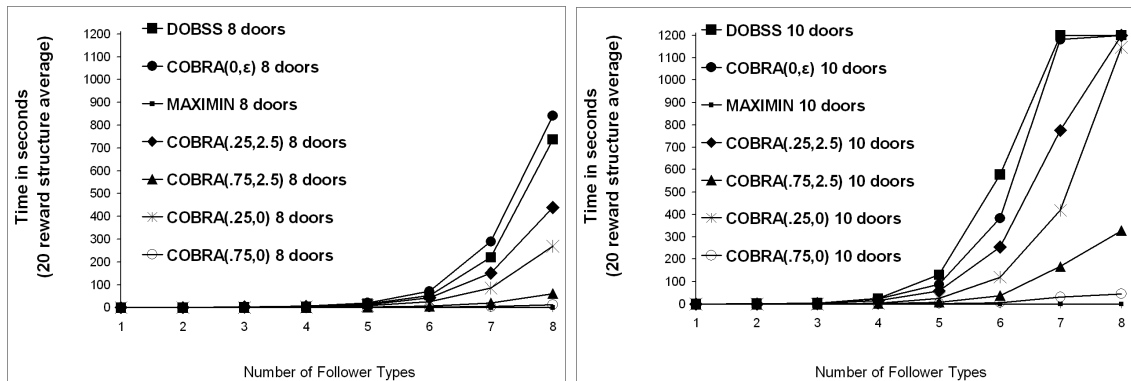


Figure 4.7: Comparing runtimes

Chapter 5: MATCH Algorithm

In Chapter 4 I demonstrated the value of addressing bounded rationality and observational uncertainty in $\text{COBRA}(\alpha, \epsilon)$. Since the inception of $\text{COBRA}(\alpha, \epsilon)$, different models have been proposed for addressing the bounded rationality of human adversaries [Pita et al., 2010; Yang et al., 2011]; however, in security game settings, the approach known as BRQR [Yang et al., 2011] presented in Section 2.5 has emerged as the leading approach under unlimited (i.e., perfect) observation conditions.

While $\text{COBRA}(0, \epsilon)$ avoids the difficult task of predicting the human-response function by instead taking an approach based on robust optimization [Aghassi and Bertsimas, 2006] that protects against worst-case deviations within an ϵ -bound, its robustness feature does not account for any deviations beyond the ϵ -bound. Thus, for deviations larger than ϵ , the defender (leader) can still suffer arbitrarily large degradations in her expected value. BRQR is a model that attempts to model all potential deviations under perfect observation, which may be why it has been shown to outperform $\text{COBRA}(0, \epsilon)$.

To that end, I significantly modify the approach proposed in $\text{COBRA}(0, \epsilon)$ based on the standard worst-case assumption of robust optimization and, instead, bound the defender's loss for a

potential deviation by the human attacker based on the degree of the deviation from the expected-value-maximizing strategy. This is done in an effort to also address all potential deviations as in BRQR, but still avoid the complex task of modeling human decision making by relying on a robust approach. This new algorithm, MATCH, provides three key benefits: (i) it provides significant runtime benefits over BRQR; (ii) it strongly couples the adversary's and defender's performance, robustly guarding against potential deviations by human adversaries and avoiding situations where minor deviations (i.e., deviations that result in minor losses in expected value) by the adversary may result in large losses for the defender; (iii) it avoids the dilemma of creating an accurate opponent model. I refer to this new type of optimization as graduated optimization and show in Section 5.1 that it lies within a space of robustness between MAXIMIN and the standard game-theoretic optimal solution.

To evaluate the advantages of this new approach, I make the most comprehensive investigation to date. Specifically, I examine 104 security settings where I take the four recommended security settings from Yang et al. [2011] and I also intelligently select 100 additional payoff structures, which I will describe in detail in Section 5.2. Furthermore, in Section 5.2 I defend why I believe future experimental setups should follow this particular experimental setup as a guideline. I test the 104 security game settings against 363 human subjects playing 8823 games in total to compare the performance of MATCH against BRQR. The results reveal that MATCH performs as well as or better than BRQR against human adversaries in over 90% of the settings tested and in Section 5.3 I give an analysis of these results.

While MATCH provides a number of benefits under unlimited observation, it is not designed to address limited observation conditions as in COBRA(α , ϵ). Furthermore, MATCH is not as robust to minor deviations, which may be more likely, as COBRA(α , ϵ) is. In the future it may

be beneficial to combine the two unique robust optimization ideas and the concept of anchoring biases into a single unified algorithm designed to be more robust for all observation conditions.

5.1 MATCH Algorithm

The key concept behind the MATCH algorithm is the new idea of graduated robust optimization. Whereas standard robust optimization robustly guards against a worst-case outcome within some error bound, MATCH assumes an expected-value-maximizing outcome on behalf of the attacker, but constrains the impact of deviations depending on the magnitude of the deviation. Specifically, the defender's loss for a potential deviation by the attacker is bounded based on the distance of that deviation from the expected-value-maximizing strategy. MATCH is specifically designed for security games, which I previously defined in Chapter 2 Section 2.6 and uses the same notation. However, in MATCH the following additional restrictions are made on the payoffs of the defender and attacker for covered and uncovered targets:

$$U_{\Theta}^u : T \rightarrow (-\infty, 0) \quad (5.1)$$

$$U_{\Theta}^c : T \rightarrow (0, \infty) \quad (5.2)$$

$$U_{\Psi}^u : T \rightarrow (0, \infty) \quad (5.3)$$

$$U_{\Psi}^c : T \rightarrow (-\infty, 0) \quad (5.4)$$

This maintains the restrictions of ΔU_{Θ} and ΔU_{Ψ} respectively and is more in line with real-world security domains where an attack creates a negative impact for the defender and a prevented attack a negative impact for the attacker. Additionally, I define $\beta \in [0, \infty)$ and restrict the attacker

to choosing a pure strategy $\psi_i \in \Sigma_\Psi$. The justification for only considering attacker pure strategies is the same as previously described for DOBSS in Chapter 2 Section 2.4. In the MATCH MILP the defender's goal is to maximize her expected value which I represent as V and the defender solves the following:

$$\begin{aligned} \max \quad & V \\ \text{s.t.} \quad & \sum_{i \in T} c_i = K \end{aligned} \tag{5.5}$$

$$c_i \in [0 \dots 1] \quad \forall i \in T \tag{5.6}$$

$$\psi = \arg \max_{\psi_i \in \Sigma_\Psi} U_\Psi(c, \psi_i) \tag{5.7}$$

$$V \leq U_\Theta(c, \psi) \tag{5.8}$$

$$\beta \cdot (U_\Psi(c, \psi) - U_\Psi(c, \hat{\psi})) \geq V - U_\Theta(c, \hat{\psi}) \quad \forall \hat{\psi} \in \Sigma_\Psi, c_{\hat{\psi}} < 1 \tag{5.9}$$

Constraints 5.5 and 5.6 ensure that the defender utilizes all her resources and that no target has more than 1 resource assigned to it. Constraint 5.7 ensures that the attacker chooses a target that maximizes his expected value. Constraint 5.8 ensures that the defender obtains the corresponding expected value (V) to the attacker's optimal strategy. Constraint 5.9 is the most crucial portion of the formulation. The left portion calculates the attacker loss in expected value for a deviation from the optimal strategy. The right hand side constrains the defender loss in expected value for this deviation by the attacker to be no more than a factor of β times the loss the attacker receives. For example, if the defender does not want to lose any more than twice what the attacker loses for a potential deviation, then we can set $\beta = 2$. If the defender does not want to lose any more than

half what the attacker loses, then we can set $\beta = .5$. This provides a direct trade-off between the defender's maximum expected value for the attacker's optimal strategy and additional protection on potential weaknesses for deviations.

MATCH addresses important issues in both COBRA and BRQR. *A fundamental property of MATCH is that it does not rely on some complex non-linear non-convex optimization problem (e.g., as in BRQR or other approaches such as RPT Yang et al. [2011]).* Indeed, its power is in its perceived simplicity, which not only means it is simple to implement, but it is orders of magnitude faster than its competitors including BRQR. In addition, similar to BRQR, it allows for a more gradual defense against deviations as opposed to a hard cutoff point as in the ϵ -rationality approach of COBRA. However, MATCH avoids the challenge of creating an accurate opponent model of human decision making by relying instead on a form of robust optimization. Nonetheless, MATCH still faces one crucial consideration, which is a trade-off between robustness and defender expected value.

The key difference between MATCH and BRQR is that BRQR attempts to model human decision making; but if this model is inaccurate, the defender's performance suffers. MATCH in contrast bypasses modeling of the human-decision-making process; it instead directly focuses on how much maximum expected value a defender is willing to trade off to protect against the human attacker's potential deviations from the rational strategy. While the β -parameter can be adjusted, in the experimental sections I will consistently keep $\beta = 1$ and show that even with this flat setting without any tuning, MATCH outperforms BRQR with careful tuning of λ . Furthermore, the performance of MATCH might also be enhanced with alternative β -settings, however, finding an appropriate procedure for estimating the β -parameter is left for future work.

Proposition 1. *If $\beta = 0$, then MATCH maximizes the worst-case outcome for the defender.*

Proof. If $\beta = 0$ it follows that Constraint 5.9 becomes $V \leq U_{\Theta}(c, \hat{\psi}) \quad \forall \hat{\psi} \in \Sigma_{\Psi}, c_{\hat{\psi}} < 1$. If $c_{\hat{\psi}} < 1 \quad \forall \hat{\psi} \in \Sigma_{\Psi}$, by definition Constraints 5.8 and 5.9 maximize the leader's minimum expected value since $V \leq U_{\Theta}(c, \psi) \quad \forall \psi \in \Sigma_{\Psi}$. If $\exists c_{\hat{\psi}} = 1$ then I will show that $U_{\Theta}(c, \hat{\psi}) \leq U_{\Theta}(c, \psi') \quad \forall \psi' \in \Sigma_{\Psi}, c_{\psi'} < 1$, guaranteeing that $U_{\Theta}(c, \hat{\psi})$ is the best worst-case bound by definition of a security game since $c_{\hat{\psi}}$ cannot be increased further. If there exists more than one target that is fully covered (i.e., $c_{\hat{\psi}} = 1$) it is only necessary to consider the target that gives the defender the least expected value among them. Consider (c, ψ) an optimal solution for MATCH with $\beta = 0$. Let $c_{\hat{\psi}} = 1$ and assume $\exists \psi' \in \Sigma_{\Psi} : U_{\Theta}(c, \psi') < U_{\Theta}(c, \hat{\psi}), c_{\psi'} < 1$. It follows from Constraint 5.9 that $V \leq U_{\Theta}(c, \psi')$ (here it may be the case that $\psi' = \psi$). By definition of a security game, the defender's expected value could be improved by increasing the value of $c_{\psi'}$ and this could be accomplished by directly trading probability from $c_{\hat{\psi}}$ to $c_{\psi'}$ at least until $U_{\Theta}(c, \hat{\psi}) = U_{\Theta}(c, \psi')$, a contradiction since (c, ψ) is an optimal solution. \square

Proposition 2. *If β is sufficiently large, then a coverage vector and an attacker strategy (c, ψ) that is optimal for the MATCH MILP corresponds to at least one SSE of the game¹.*

Proof. As stated in Definition 1, in order for (c, ψ) to be a SSE they must meet three defined criteria: (i) the leader plays a best response, (ii) the follower plays a best response, (iii) the follower breaks ties optimally for the leader. If β is sufficiently large then Constraint 5.9 is now trivially satisfied for any target not in the attack set ($\pi(c)$), effectively removing it. For targets in the attack set, I will show that in any optimal solution the MATCH MILP will force the attacker (follower) to break ties in the defender's (leader's) favor. Then I will show that the MATCH MILP forces the defender and attacker to play mutually best responses.

¹In this case Constraint 5.9 is effectively removed making the MATCH MILP exactly equivalent to an existing MILP known as ERASER which has previously been shown to compute optimal solutions that correspond to at least one SSE of the game[Kiekintveld et al., 2009]

Consider a solution with $\psi \in \pi(c), \psi \neq t^*$ where t^* is as defined in Section 2.6. It follows that $U_{\Psi}(c, \psi) = U_{\Psi}(c, t^*), U_{\Theta}(c, \psi) \leq U_{\Theta}(c, t^*),$ and $V \leq U_{\Theta}(c, \psi)$. For target t^* , Constraint 5.9 now becomes $0 \geq V - U_{\Theta}(c, t^*)$. If $U_{\Theta}(c, t^*) > V$ then the constraint is satisfied, however, the defender could do better if the attacker deviates from ψ to t^* . By definition of a security game, the defender can induce this deviation by removing an arbitrarily small coverage from c_{t^*} and adding it to c_{ψ} , thus inducing the favorable SSE by selecting a strategy arbitrarily close to the equilibrium [von Stengel and Zamir, 2004]. This change would create a coverage vector $c^* \approx c$ with an attack set that would only contain t^* and the defender would receive utility $V \leq U_{\Theta}(c^*, t^*)$. To verify, by removing an arbitrarily small coverage probability from c_{t^*} , it has forced $U_{\Psi}(c, t^*) > U_{\Psi}(c, t) \quad \forall t \in T, t \neq t^*$, including any other target previously in the attack set since no coverage probability was removed from other targets. In the opposite direction, if $\exists \hat{\psi} \in \pi(c), U_{\Theta}(c, \hat{\psi}) < U_{\Theta}(c, \psi)$ then Constraint 5.9 is no longer satisfied. In order to satisfy this constraint the MILP would force the attacker to choose the less favorable target ($\hat{\psi}$) and by the same logic the favorable outcome could be induced, removing the unfavorable outcome from the set.

So far I have shown that the MATCH MILP forces the attacker to break ties in the defender's favor. It remains to be shown that the MATCH MILP forces the defender and attacker to play best responses as defined in a SSE. In MATCH, Constraint 5.8 defines the defender's expected value (V) contingent on the target attacked (ψ). The constraint places an upper bound of $U_{\Theta}(c, \psi)$ on V . Since the objective maximizes V for any optimal solution $V = U_{\Theta}(c, \psi)$. This also implies that c is maximal, given ψ for any optimal solution, since V is maximized. In a similar way, Constraint 5.7 forces the attacker to select a target that maximizes his expected value given a coverage vector

c . Taken together, the objective and Constraints 5.7, 5.8 and 5.9 (for tie breaking) imply that (c, ψ) are mutual best-responses in any optimal solution.

While I have shown that (c, ψ) are mutual best responses, it remains to be shown that c corresponds to a mixed strategy $x \in X$ for the defender and that the attacker's Stackelberg strategy F_Ψ can be constructed. As previously stated, it has been shown that the coverage vector $c \in C$ can be converted into an equivalent mixed strategy $x \in X$ [Kiekintveld et al., 2009]. While c corresponds to a mixed strategy for the defender, ψ is an incomplete description of the attacker's Stackelberg strategy F_Ψ ; it does not specify choices for any coverage other than c . I will show that the constraints of the MILP imply the existence of a function F_Ψ extending ψ such that c and F_Ψ satisfy the criteria of a SSE. While I have shown that c and ψ are mutual best-responses for an optimal MILP solution, it remains to describe the attacker's behavior for any other feasible coverage vectors $c' \neq c$. Let $\psi' = t^* \in \pi(c')$ be a target in the attack set for c' with maximal expected value for the defender. By construction, ψ' is feasible in the MILP and satisfies criteria 2 for a SSE. Based on the selection of $\psi' = t^* \in \pi(c')$, remember that Constraint 5.9 will ensure that $U_\Theta(c', t) = U_\Theta(c', t^*) \quad \forall t \in \pi(c')$, satisfying criteria 3 for a SSE. Since (c', ψ') is a feasible solution in the MILP, $U_\Theta(c', \psi') \leq U_\Theta(c, \psi)$ since (c, ψ) is optimal for the MILP. Let F_Ψ be a function constructed using this method for every possible $c' \neq c$. c is a best-response to F_Ψ since $U_\Theta(c', \psi') \leq U_\Theta(c, \psi)$, satisfying the first criteria of a SSE. □

5.2 Experiment Purpose, Design, and Results

5.2.1 Purpose of this Study

I sought to investigate the performance of BRQR and DOBSS against MATCH under perfect observation. Here, performance is measured by the average expected value obtained by a security force against the decisions of human adversaries. In these experiments I only examine one crucial variable of real-world domains, which is the reward structure, while the other variables are held constant. For BRQR I fixed $\lambda = .75$ and for MATCH I fixed $\beta = 1.0$. While I discuss estimating appropriate λ -settings in Section 5.4, my choice of $\lambda = .75$ was inspired by the original estimate made by Yang et al. [2011]. I chose $\beta = 1$ to constrain the defender losses to be no worse than the attacker losses for deviations. In Section 5.2.2.2 I will describe how the reward structures were chosen.

The goal of MATCH was to increase the performance of a security force against human adversaries by addressing the bounded rationality that humans may exhibit under perfect observation conditions. To that end, experiments were set up where human subjects would play as followers (adversaries) against each strategy under different reward structures given perfect observation. It is not possible to prove optimality against human adversaries who may deviate from the expected optimal responses and thus I rely on empirical validation through experimentation. In addition to examining the performance of BRQR with MATCH, I also examine the runtime performance of MATCH against BRQR.

5.2.2 Experimental Design

Similar to the experimental design in Chapter 4, these experiments are inspired by the real-world security domain at LAX [Pita et al., 2008] described previously. Here, three guards – jointly acting as the defender – guard eight gates, and an individual human subject acts as a single attacker who will choose one of the eight possible gates. Again, the 8 gates model the 8 terminals found at LAX. The interface can be seen in Figure 5.1

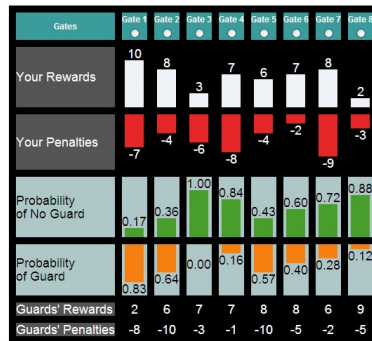


Figure 5.1: Game Interface

Each subject was modeled as a single follower type defined by the reward structure they were given. In order to simulate the Stackelberg setting, subjects were presented with the following information before they chose a gate: (i) the subject's reward and penalty for each gate; (ii) the defender strategy (i.e., the probability distribution of the guards over the 8 gates); (iii) the guard's reward and penalty for each gate. This can be seen in Figure 5.1. Again, I use this Stackelberg framework because in real-world scenarios an attacker can conduct extensive surveillance of his potential target and the corresponding defensive strategy before choosing to attack, which would allow him to learn this information.

In each game instance, the guards would choose 3 gates based on their strategy and the subject's goal was to choose the gate that would maximize expected value given the defender's strategy. Each of the 8 gates would have a different reward and penalty associated with it for both the subjects as well as the guards. For instance, as shown in Figure 5.1, gate 4 has a reward of 7 and a penalty of -8 for the subject and a reward of 7 and a penalty of -1 for the guard. If a subject chose a gate that a guard was guarding, the subject would incur the subject penalty for that gate and the guard would receive the guard reward for that gate, else vice-versa. Going back to the previous example, if the subject chose gate 4 and a guard was guarding that gate then the subject would receive -8 and the guard would receive 7. This setup led to a Stackelberg game with $\binom{8}{3} = 56$ leader actions, and 8 follower actions.

5.2.2.1 Participants

These experiments were run in Amazon Mechanical Turk with the requirement that workers were from the United States. Outside of the qualification that the workers were from the United States, the workers were anonymous except for an identifying worker number. This worker number was tracked in order to insure a worker could not participate in more than one of the following experiments. There are three different experimental evaluations that were conducted in this study. In the first experimental evaluation there were 69 unique (i.e., did not participate in any of the other experiments or more than once in this experiment) anonymous Amazon Mechanical Turk workers from the United States. In the second experimental evaluation there were 253 unique anonymous Amazon Mechanical Turk workers from the United States. In the third experimental evaluation there were 41 unique anonymous Amazon Mechanical Turk workers from the United States.

5.2.2.2 Reward Structure

Given this experimental setup, I ran experiments in two sets of reward structures. I first explored four reward structures proposed by Yang et al. [2011], which were chosen to be the most representative of the entire payoff structure space for security games based on metrics proposed by Yang et al. [2011]. These reward structures can be found in the Appendix under Section B, Tables B.5-B.8. However, I found that in these particular reward structures the strategies produced by BRQR and MATCH were highly similar. Thus, to more fully compare the performance of BRQR and MATCH, for the second set of experiments I systematically chose 100 new reward structures based on covariant games in GAMUT [Nudelman et al., 2004].

For Covariant games, GAMUT creates a game for a given number of players with payoffs distributed normally(0,1) with covariance $r \in [-1, 1]$, which determines a covariance between player rewards. Specifically, when $r = -1$ the game is zero-sum between players and when $r = 1$ the game is perfectly cooperative. I slightly modified the code in GAMUT to restrict the resulting payoffs to meet the criteria of a security game. That is, the payoffs fall within a specified range of $[-10, -1]$ for penalties and $[1, 10]$ for rewards. I chose covariant games to generate payoff structures because they are naturally able to capture the adversarial nature of security games.

To select the 100 new reward structures I first generated 1000 reward structures with the r parameter ranging from -.9 to 0 by .1 increments (i.e., 100 games for each setting of r). I chose 0 as the boundary because, in an adversarial security setting, it does not make sense that the reward and penalty vectors would be positively correlated. In an attempt to examine how different the MATCH and BRQR algorithms are overall, for each structure I computed the 1-norm distance between MATCH and BRQR where, as I explained previously, I set $\lambda = .75$ and $\beta = 1$.

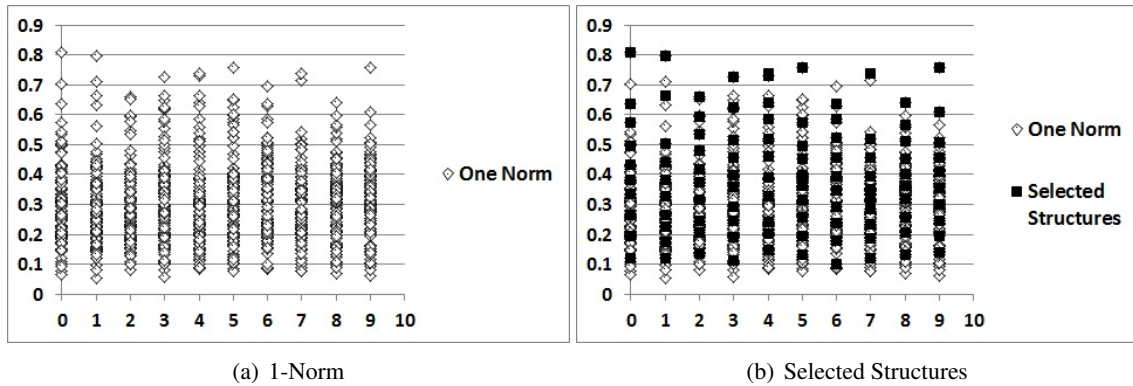


Figure 5.2: 1-Norm Scatter Plots

In Figure 5.2(a) I present the scatter plot for these 1000 reward structures. For readability, on the x-axis I display the setting of r from 0 to $-.9$ as 0 to 9 (i.e., on the x-axis 3 represents $-.3$). On the y-axis I display the 1-norm value. These scatter plots show that there is a wide range of possibilities for the difference between MATCH and BRQR strategies. While extreme differences don't occur as frequently, they do not appear to be completely rare. Given these 1000 structures, I attempted to select 100 reward structures that would best cover the possible space based on the 1-norm values. I present the 100 reward structures selected within the scatter plot in Figure 5.2(b) and the actual reward structures can be found in the Appendix under Section B, Tables B.9-B.108.

I believe this procedure for selecting reward structures is superior to previous procedures [Pita et al., 2010; Yang et al., 2011] for three critical reasons: i) by examining the 1-norm distance I can explore a spectrum of reward structures where the strategies produced are most different (high 1-norm) to where they are most similar (low 1-norm); ii) by utilizing covariant games, I can control the correlation between player rewards to ensure rewards and penalties are not positively correlated where previous experiments have ignored this crucial issue [Yang et al., 2011; Pita et al., 2010]; and iii) previous experimental results may give a distorted view of the

overall performance of an algorithm compared to other algorithms since they look at such a narrow portion of the entire security game space (i.e., 4 to 10 potential settings versus over 100).

5.2.2.3 Experimental Procedure

All of these experiments were run in Amazon Mechanical Turk and participants were paid a base amount of US \$1.50 for participating. For the first experiment, I tested the mixed strategies generated by DOBSS, BRQR and MATCH in the first set of reward structures (i.e., the four structures proposed by Yang et al. [2011]). I used the original λ estimate made by Yang et al. [2011] (i.e., $\lambda = .76$) and $\beta = 1$ to constrain the defender's losses to be no worse than the attacker's losses for MATCH. Each of the 12 game settings (four reward structures and three algorithms) were played by 40 subjects (i.e., in total there were 480 total trials).

The 12 potential game settings were broken into two separate groups of 6 game settings. The first group of 6 game settings consisted of all combinations of game settings for reward structures 5 and 6 (i.e., 2 reward structures and three algorithms). Similarly, the second group consisted of the same for reward structures 7 and 8. All experiments were first conducted for group 1 with 40 subjects and then group 2 with 40 different subjects. The following procedure was used for both groups.

First, to ensure that subjects were not choosing gates arbitrarily, I introduced two obvious games where a gate with the highest reward and lowest penalty possible had the lowest probability (5%) of being covered. These games and corresponding strategies can be found in the Appendix under Section G.2.1, Tables G.1 and G.2. If subjects did not choose this gate in these two games their results were removed from the final set. This led to less than 40 subjects in the final data sets presented, which accounts for the number of participants presented in Section 5.2.2.1.

In order to mitigate the order effect on subject responses, a total of 8 different orderings of the 8 game settings (6 original settings and 2 dummy game settings) were generated using Latin Square design. Every ordering contained each of the 8 game settings exactly once, and each game setting appeared exactly once in each of the 8 positions across all 8 orderings. The order played by each subject was drawn uniformly randomly from the 8 possible orderings. Before beginning the game, subjects were given a brief tutorial about how to play to ensure that they understood the general game play. This tutorial can be found in the Appendix under Section G.3. After beginning, subjects were given an unlimited amount of time to make a decision on which gate to choose for each game.

As stated earlier, subjects were paid a base amount of US \$1.50 for participating in this study. However, to further motivate the subjects, they were allowed to earn additional bonus money based on their performance in the game. For each reward point earned or lost, a subject would receive or lose an additional US \$0.15 from their net money. Before playing, subjects were informed that only 5 of their games would be selected from the 8 games played to determine the actual bonus payment. Since subjects were not aware which games would be chosen, they would have incentive to perform the best they could in each game and they were given immediate feedback at the end of each game. Subjects were paid their final earnings, but were not required to pay money back if their net points were less than zero.

For the second set of experiments, I tested the mixed strategies generated by BRQR and MATCH in the second set of reward structures (i.e., the 100 structures chosen using Gamut). I chose to omit DOBSS in these experiments since it has been previously shown that DOBSS performs poorly against human adversaries [Yang et al., 2011], and the previous experiment along with the experiments with COBRA also demonstrate this result. This lead to a total of 200

possible game settings (100 reward structures and two algorithms). The exact same procedure described for the first experiment was used for this experiment, except the 200 potential game settings were grouped differently.

To avoid boredom in the subjects, I limited the number of games they would have to play by separating the reward structures into the following groups played by the number of subjects indicated: (i) Reward structures 9-13 (I start at 9 to account for the previous 9 reward structures) [30 participants (25 after removal)], (ii) Reward structures 14-25 [40 participants (33 after removal)], (iii) Reward structures 26-40 [40 participants (37 after removal)], (iv) Reward structures 41-57 [45 participants (40 after removal)], (v) Reward structures 58-74 [40 participants (37 after removal)], (vi) Reward structures 75-91 [45 participants (42 after removal)], and (vii) Reward structures 92-108 [40 participants (39 after removal)]. Subjects would play against both BRQR and MATCH for a given group of reward structures. The only difference between this experimental setup and that described for the first experiment was the number of games played and the corresponding Latin Square. For example, subjects who played reward structures 14-25 played a total of 26 game settings (24 game settings from the reward structures and 2 from the dummy games), which lead to 26 different orderings of the 26 game settings. As before, every ordering contained each of the 26 game settings exactly once, and each game setting appeared exactly once in each of the 26 positions across all 26 orderings. The order played by each subject was drawn uniformly randomly from the 26 possible orderings.

For the third set of experiments I tested the mixed strategies generated by BRQR and MATCH. Again, I chose to omit DOBSS for the reasons explained in the second experiment. The same procedure was used as in the previous two experiments. I leave discussion on which game settings were tested for Section 5.4 since it relies on the results of the second experiment.

In the following, I will first present the results for the reward structures proposed by Yang et al. [2011], then the results for the newly selected structures, and finally I will give an analysis of these results. I evaluate the statistical significance for all results using the bootstrap-t method [Wilcox, 2003] used by Yang et al. [2011] previously, which is described in the Appendix under Section A.

5.2.3 Results for Original Structures

I present the results of these experiments in Figure 5.3. In total, 36 subjects played against reward structures 5 and 6 while 33 played against reward structures 7 and 8 (after removal of subjects who failed the obvious games). In Figure 5.3, the y-axis represents the average defender expected value over all the choices made by each individual subject against a particular strategy. For example, examining reward structure 5 in Figure 5.3, we see that the defender received -0.29 on average against human subjects if using the strategy generated by BRQR.

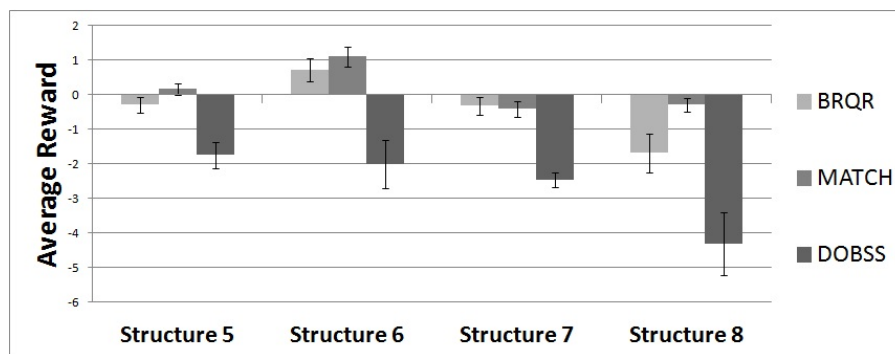


Figure 5.3: Original reward structures

In reward structures 5 and 8 MATCH is statistically significantly better than BRQR and DOBSS ($p \leq .028$ and $p \leq .004$ respectively). In reward structures 6 and 7 BRQR and MATCH are not statistically significantly better than each other ($p = .15$ and $p = .392$ respectively),

but both are statistically significantly better than DOBSS ($p = 0$). However, in general across all 4 payoff structures, MATCH and BRQR create highly similar strategies (i.e., the probability difference on any particular gate is relatively low [$\leq 12.6\%$]). Regardless, even with the similar strategies, MATCH still outperforms BRQR with statistical significance in two of the four reward structures and does at least as well in the other two. Based on these results I can conclude that there exists conditions where MATCH is the superior algorithm to BRQR and once again accounting for human adversaries is crucial since both algorithms significantly outperform DOBSS.

5.2.4 Results for New Reward Structures

While the results from the first experiment were promising, I wanted to take the most extensive look to date at a large space of potential security game settings, examining 100 potential reward structures compared to 10 or less in previous experiments [Yang et al., 2011; Pita et al., 2010]. I present an overview of the results from these experiments in Table 5.1. Here, I show the number of settings where MATCH won with statistical significance, both strategies were approximately equivalent (i.e., neither strategy won with statistical significance), and BRQR won with statistical significance. In 42 of the 100 reward structures MATCH outperformed BRQR with statistical significance and in an additional 52 of the 100 reward structures MATCH did at least as well as BRQR given that neither strategy won with statistical significance. These results combined with the previous experiment show MATCH performing at least as well as or outperforming BRQR in 98 out of 104 potential security settings. In Section 5.3 I will give further analysis of these results.

	MATCH	Draw	BRQR
$\alpha = .05$	42	52	6

Table 5.1: Overview of Results

	MATCH	Draw	BRQR	TOTAL
Rejected	40	40	3	83
Not Rejected	2	12	3	17

Table 5.2: Pearson Chi-squared Results

5.3 Analysis

The QR model is a well-established solution concept and so an important question to address is whether BRQR was actually an accurate model of human decision making in these security settings. To determine whether BRQR is accurate, in each reward structure I run a Pearson's chi-squared goodness of fit test [Greenwood and Nikulin, 1996] on the predicted distribution of attacker choices against the observed attacker choices for the subjects. I present the results in Table 5.2. In the first three columns I denote reward structures where MATCH won with significance, neither strategy won with significance, and BRQR won with significance. In the last column I give the overall result for all 100 reward structures. In the rows I denote whether Pearson's chi-squared goodness of fit test rejected the null hypothesis that the observed distribution of choices could have been drawn from the expected distribution of choices ($\alpha = .05$).

My first observation is that in 83% of all reward structures tested the model proposed by BRQR did not fit the data observed. This is a significant number and suggests that perhaps BRQR is not a good model of human decision making in security games. However, it is possible that this result is due to a poor estimation of the λ -parameter for these particular security settings and so in Section 5.4 I will re-estimate the λ -parameter based on the observed data and run additional experiments for a key subset of the reward structures. Even so, in real-world security settings it

may be even more difficult to appropriately estimate λ since data can often be sparse or noisy, and the problem instances can be much larger and more complex.

The fact that, for this set of results, BRQR does not provide a good fit for the data observed in general is one potential explanation for why MATCH is outperforming BRQR in the majority of the security settings. BRQR attempts to exploit an assumed model of the human attacker and if the attacker deviates from that model in a significant way it can severely impact the performance of the defender. For instance, if BRQR assumes that an attacker is not likely to attack a certain target it will provide minimal coverage for that target. If, however, a large number of attackers choose this target, contrary to the prediction, it can have severe effects on the defender's average expected value. An inaccurate model of human decision making can lead to severe consequences in security domains. This is one of the key advantages of MATCH since it does not assume any decision-making model on behalf of the attacker and specifically bounds the impact of such potential deviations.

My second observation is that of the 17 structures where BRQR was potentially a good fit of human decision making, it only outperformed MATCH in 3. Thus, of the six cases where BRQR won with statistical significance, only three of the cases can potentially be justified by accurate opponent modeling. These results show that even a decent model of human decision making may not be sufficient enough and an approach based on robust optimization is potentially a strong alternative to modeling human-decision-making processes. It may be necessary to have a highly accurate model of human decision making before it becomes sufficient in security settings.

A second important question is, given the space of possible reward structures, is it possible to determine when MATCH or BRQR will likely perform better. In Figure 5.4, I present where the results for the 100 structures appear in the scatter plot of 1000 structures. In the cases where

neither MATCH nor BRQR won with statistical significance I present which strategy had the higher average defender expected value. While this does not imply that the strategy is better in these cases, this was done to see if it would provide some insight into what portion of the payoff structure space MATCH or BRQR performed better. Here, it is evident that the results are diverse within the potential space of reward structures and so without further investigation I cannot determine where BRQR will likely be better than MATCH overall. However, these results support my earlier argument that when doing experimental analysis in security settings, examining few potential settings can give misleading results. For example, for a small sample of potential reward structures chosen throughout the potential space, it is possible I could have chosen only structures where BRQR outperformed MATCH, which would be contrary to what I have been able to show in my results. Thus, I have rightly raised the standard for future experimental investigations of new potential algorithms.

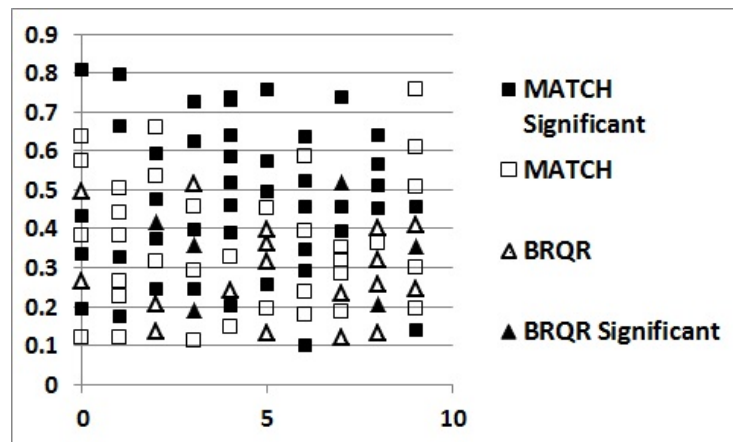


Figure 5.4: Scatter Plot of Results

5.4 λ -Re-estimation

As suggested in my analysis, in order to confirm whether BRQR is actually a poor predictor of human decision making I am required to examine BRQR with appropriately estimated λ -values. To focus my analysis I selected three groups of five reward structures from the 42 reward structures where MATCH outperformed BRQR with statistical significance as follows: (i) the five reward structures where BRQR and MATCH had the most significant strategy difference averaged over the 1-norm, 2-norm, infinite-norm, and KL distances; (ii) the five reward structures where BRQR and MATCH had the least difference in average expected value; and (iii) the five reward structures where BRQR and MATCH had the highest difference in average expected value. For these experiments I will refer to these as reward structures 1 through 15. To re-estimate the λ -parameter I used the Maximum Likelihood Estimation procedure proposed by Yang et al. [2011] using the data from the previous experiments in each of the 15 reward structures yielding 15 new λ -values. In Table 5.3 I present the new λ -estimates along with the 1-norm distance between the strategy produced by the original λ -setting ($\lambda = .75$) and the re-estimated λ -setting.

My first observation is that the value of λ is largely dependent on the reward structure implying that for each potential security domain the defender would be required to make a new estimate. As stated previously, estimating λ can already be difficult in real-world settings where data may be sparse or noisy. This problem is further exacerbated since I have now shown that data cannot likely be pooled from different settings. This is an additional advantage of the approach in MATCH since the level of robustness is not dependent on the reward structure. That is, once the level of robustness has been decided (i.e., β -setting) it is consistent across all reward structures where a λ -setting is not equivalently accurate for all reward structures.

Structure:	1	2	3	4	5	6	7	8
λ :	.18	.71	.25	1.14	.01	1.39	1.09	.67
1-norm:	.376	.012	.384	.177	3.10	.221	.110	.050
Structure:	9	10	11	12	13	14	15	
λ :	.84	.48	.15	.43	.55	.23	.42	
1-norm	.018	.069	.424	.396	.127	.244	.356	

Table 5.3: New λ -estimates

My second observation is that the actual impact of altering the λ -parameter varies significantly depending on the reward structure. For example, in structure 10 I vary λ from .75 to .48 and see only a 1-norm difference of .069 while in structure 12 I reduce λ to .43 and see a 1-norm difference of .396 (i.e., there is a 40% difference of probability across targets in one case and only a 7% difference across targets in another case). This once again demonstrates the difficulty in appropriately estimating λ since minor changes can lead to significant differences in some reward structures.

I had 41 subjects play against both MATCH ($\beta = 1$) and BRQR with newly estimated λ -values in all 15 reward structures using the procedure outlined in Section 5.2.2.3. I present the results in Figure 5.5. In these results, MATCH remained statistically significantly better in 8 of the 15 reward structures (structures 1-5, 11, 13, and 15) and neither strategy was statistically significantly better in the remaining 7. Additionally, I ran Pearson's chi-squared goodness of fit test and found that, even after re-estimation, the null hypothesis that the observed choice distribution could have been drawn from the predicted choice distribution was rejected in all 15 cases. Thus, even if I obtain tailored λ estimates, MATCH continues to perform as well as or outperform BRQR.

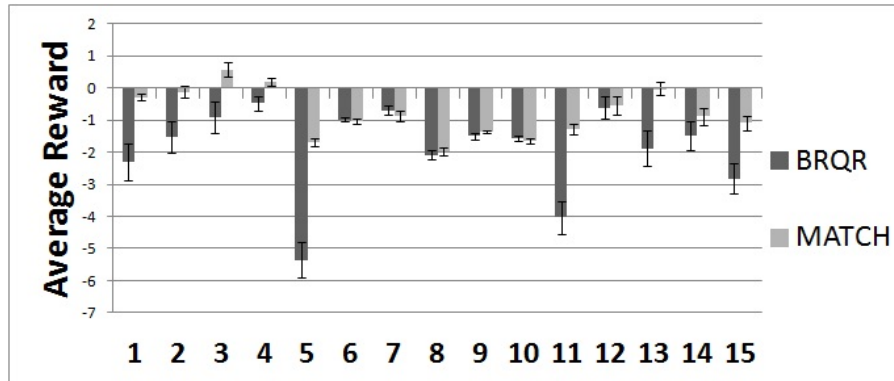
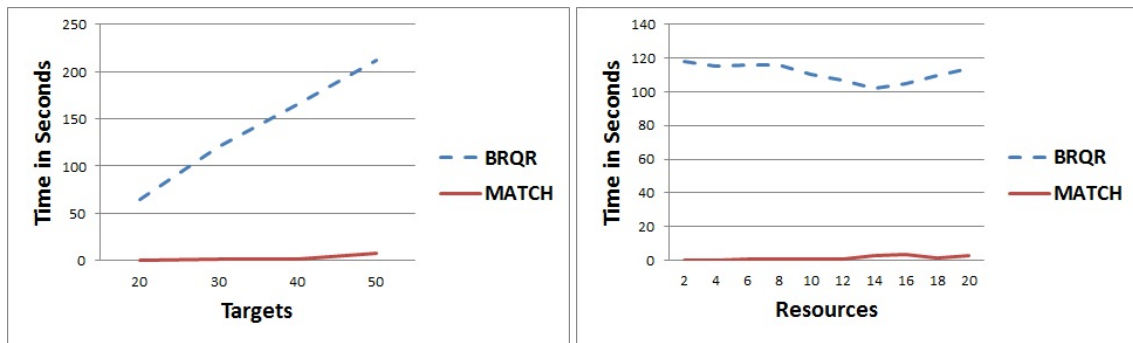


Figure 5.5: Re-estimated Reward Structures

5.5 Runtime Results

In Figure 5.6 I present runtime results for BRQR versus MATCH. In Figure 5.5, the number of resources are fixed at 10 and on the x-axis I vary the number of targets from 10 to 50. On the y-axis I present the runtime in seconds averaged over 20 randomly generated payoff structures. In Figure 5.5, the number of targets are fixed at 30 and on the x-axis I vary the number of resources from 2 to 20. Once again on the y-axis I present the runtime in seconds averaged over 20 randomly generated payoff structures. Experiments were run using CPLEX 8.1 on an Intel(R) Core(TM) i5-2450M CPU 2.50GHz processor with 6 GB RAM.

These results show that MATCH provides orders of magnitude speedup over BRQR further demonstrating the benefits of such an approach. The reason for this runtime improvement is that BRQR requires the solution to a non-linear and non-convex objective function in its most general form. In fact, because of the complexity of the objective function, BRQR is only a heuristic solution for solving the objective. MATCH on the other hand is a mixed-integer linear program, which can be solved with standard packages. These results demonstrate that MATCH significantly improves runtime costs for the proposed enhancements to deal with bounded rationality.



(a) Increasing Targets

(b) Increasing Resources

Figure 5.6: Runtime results

Based on these experimental results I have demonstrated five fundamental contributions of the MATCH algorithm: (i) I develop an approach to addressing human adversaries based on robust optimization rather than relying on finding appropriate models of human decision-making; (ii) I introduce a systematic way to generate meaningful reward structures based on covariant games where previous work has simply generated completely random reward structures; (iii) I make the most comprehensive evaluation to date involving 363 human subjects playing 8823 games in 104 security game settings; (iv) I demonstrate that MATCH performs as well as or better than BRQR in over 94% of the security settings tested (42 of 104 settings with statistical significance); and (v) I demonstrate the significant runtime benefits of MATCH over BRQR. These results demonstrate the potential benefits of using an approach based on robust optimization (e.g., MATCH or COBRA) over previous algorithms that rely on creating more efficient models of human-decision making.

Chapter 6: Security Circumvention Games

There are three strict assumptions I aimed to address in this thesis. Namely, that human adversaries are perfectly rational, that they perfectly observe the leader's strategy, and that their action space is tractable. In Chapters 4 and 5 I addressed issues surrounding the assumptions of perfect rationality and perfect observability. However, the action space assumption remains to be addressed. To motivate the importance of addressing this assumption, I will use a real-world scenario faced by the United States Transportation Security Administration (TSA) that lead to my new modeling approach and an application that is under evaluation [Pita et al., 2011].

The TSA is tasked with protecting the nation's transportation systems [TSA, 2012]. These systems are often large in scale and require many personnel and security activities to protect them. One set of systems in particular is the over 400 airports [TSA, 2012]. These airports serve approximately 60 million aircraft annually [NATCA, 2012]. To protect this large transportation network, the TSA employs approximately 48,000 Transportation Security Officers [TSA, 2012]. These Security Officers are responsible for implementing security activities at each individual airport in order to provide security for the transportation network.

While many people are aware of common security activities, such as individual passenger screening, this is just one of many security layers TSA personnel implement to help prevent

potential threats [TSA, 2012]. These layers can involve hundreds of heterogeneous security activities executed by limited TSA personnel leading to a complex resource allocation challenge. Unfortunately, the TSA cannot possibly run every security activity all the time and thus must decide how to appropriately allocate its resources among the layers of security to protect against a number of potential threats. To aid the TSA in scheduling resources in a risk-based manner, I take a multi-agent game-theoretic approach. From the perspective of the underlying game-theoretic model, a crucial difference of my novel approach and previous approaches is this: I allow for both heterogeneous security activities and threats whereas the security games presented previously are only able to consider homogeneous security activities and threats, leading to a new game model called “Security Circumvention Games” (SCGs).

In conjunction with TSA subject matter experts, I developed a software system, Game-theoretic Unpredictable and Randomly Deployed Security (GUARDS), that utilizes a Stackelberg framework to aid in protecting the airport transportation network. However, the TSA’s security challenge raises two new critical issues. The first issue is in appropriately modeling the TSA’s security challenges in order to achieve the best security policies (mixed strategy). Due to the complex nature of TSA’s security challenges, traditional models of security games [Korzhyk et al., 2011] are no longer appropriate models. Specifically, the TSA’s domain has the following additional features beyond traditional security games: (i) heterogeneous security activities for each potential target; (ii) heterogeneous threats for each potential target; (iii) unique security activities for individual airports. The second issue is in efficiently solving the model I developed where, because I consider a national deployment, a special-purpose solver may not be appropriate. In fact, previous solution techniques [Jain et al., 2010; Kiekintveld et al., 2009; Jain et al., 2011b; Yin and Tambe, 2012] for traditional security games are no longer directly applicable.

To address these issues, I developed both a new formal model of security games and techniques to solve this class of games. To appropriately model the TSA's security challenges I created a novel game-theoretic model, which is referred to as Security Circumvention Games (SCGs), and cast the TSA's challenges within this model. In the creation of SCGs I provide the following contributions: (i) the ability for defenders to guard targets with more than one type of security activity (heterogeneous activities), and (ii) the ability for attackers to choose threats designed to circumvent specific security activities. Given my new model, I designed an efficient solution technique in which I create a compact representation of SCGs. This allows the application to avoid using a tailored Stackelberg solver and instead utilize a general purpose Stackelberg solver to compute solutions efficiently.

6.1 TSA Security Challenges

I now describe in detail the two major issues in potentially deploying game-theoretic randomization for airport security on a national scale, including modeling and computational challenges along with my solutions to them.

6.1.1 Modeling the TSA Resource Allocation Challenges

While I am motivated by an existing model of security games [Korzhyk et al., 2011], there are three critical aspects of the new TSA domain that raise new challenges. First, the defender now reasons over heterogeneous security activities for each potential area within an airport¹. For example, airports have ticketing areas, waiting areas, and cargo holding areas. Within each of these areas, the TSA has a number of security activities to choose from such as perimeter patrols,

¹Due to the nature of the TSA's security challenge, I will refer to targets in the TSA's domain as areas henceforth.

screening cargo, screening employees and many others. Second, given the multiple possible security activities, the defender may allocate more than one resource per area (i.e., areas are no longer covered or uncovered). Finally, the defender now considers an adversary who can execute heterogeneous attacks on an area. The TSA must reason about a large number of potential threats in each area such as chemical weapons, active shooters, and bombs. The key challenge is then how to allocate limited TSA security resources to specific activities in particular areas, taking into account an attacker's response.

To address this challenge it is necessary to create a more expressive model than outlined in security games; one that is able to reason over the numerous areas, security activities, and threats within an individual airport. We refer to this new class of security games as Security Circumvention Games (SCGs). SCGs are more expressive than traditional security games and thus can represent both traditional security games and the games I consider for the TSA. In SCGs, the TSA must choose some combination of security activities to execute within each area and the attacker must reason over both which area to attack and which method of attack to execute based on the defender's strategy. At this time I elaborate on the defender's and attacker's possible strategies.

6.1.1.1 Defender Strategies

I still denote the defender by Θ , and the set of defender's pure strategies by $\sigma_i \in \Sigma_\Theta$. The TSA is able to execute a variety of security activities, which I denote by $S = \{s_1, \dots, s_{|S|}\}$. Each security activity has two components. The first is the type of activity it represents, and the second is the area where the activity is performed. I denote the set of areas by $A = \{a_1, \dots, a_{|A|}\}$.

The defender still has $R = \{r_1, \dots, r_{|R|}\}$ resources available and thus a pure strategy, $\sigma_i \in \Sigma_\Theta$, is a subset of security activities from S with size equal to $|R|$. The TSA's task is to consider how to allocate these resources among security activities in order to provide the optimal protection to their potential areas. For example, if there are three security activities, $S = \{s_1, s_2, s_3\}$ and two resources available, one possible pure strategy for the defender is to assign these two resources to s_1 and s_3 . Given that the number of possible combinations of $|R|$ security activities at an airport can be on the order of 10^{13} or greater for the TSA, I develop a compact representation of the possible strategies that I present in Section 6.1.2. The defender's mixed strategies $\delta_\Theta \in \Delta_\Theta$ are the possible probability distributions over Σ_Θ . Similar to previous work, a mixed strategy (randomized solution) is typically the optimal strategy.

6.1.1.2 Attacker Actions

Defending an area against terrorist attacks is complicated by the diversity of the potential threats. For example, an attacker may try to use a vehicle borne explosive device, an active shooter, a suitcase bomb, and many others in any given area. Not all methods of attack would make sense in all areas. For example, using a vehicle borne explosive device in the checked baggage screening area in some airport configurations would not be a viable method of attack. I still denote the attacker by Ψ , and the set of pure strategies for the attacker is still given by $\psi_j \in \Sigma_\Psi$. Each pure strategy for the attacker corresponds to selecting a single area $a_i \in A$ to attack, and a specific mode of attack. However, given the complexity of the TSA's security challenge and that human adversaries are capable of continually evolving new threats, enumerating all the possible threats may not be practical if even possible. To address the action space assumption described

previously and avoid this difficulty, I developed a novel way to represent threats for the TSA's domain that I describe in Section 6.1.2.1.

6.1.2 Compact Representation for Efficiency

While I have developed a model that appropriately captures the TSA's security challenge, one issue with this model is that both the attacker and defender strategy spaces grow combinatorially as the number of defender security activities increases. Also, listing such a large number of potential threats would lead to extreme memory and runtime inefficiencies. Furthermore, existing solution techniques that have been developed for security games [Jain et al., 2010; Kiekintveld et al., 2009; Jain et al., 2011b; Yin and Tambe, 2012] are not directly applicable to Security Circumvention Games (SCGs).

With this in mind, I looked at an alternate approach to finding optimal solutions efficiently. Specifically, I looked at representing threats in a more intelligent manner and creating a compact representation for the defender strategy space. By utilizing both of these techniques, I achieved large reductions in runtime. I utilized DOBSS [Paruchuri et al., 2008] to solve my compact representation and avoided creating a tailored algorithm for each specific airport. At this time I will explain both how I model threats and how I achieve a compact representation of the defender's full strategy space.

6.1.2.1 Threat Modeling for TSA

While it is important to reason over all the security activities that are available to an individual airport, enumerating all of the large number of potential threats they face can lead to severe memory and runtime inefficiencies. Thus, the problem I face is how to model attack methods

in a way that limits the number of threats the game-theoretic model needs to reason over, but appropriately captures both an attacker's capabilities and his goals. In particular, I automatically generate attack methods for the adversary that capture two key goals: (i) an attacker wants to avoid the security activities that are in place, and (ii) an attacker wants to cause maximal damage with minimum cost.

In order to achieve these goals an intelligent adversary will observe security over time and design his attack method based on his observations. The attacker's plan will be designed to avoid security activities that he believes will be in place. I will refer to this as circumventing security activities. For example, imagine there is a single area with three security activities such as passenger screening, luggage screening, and perimeter patrol. In this example, the TSA only has one resource available and thus can only execute one of these activities at a time. While passenger screening may have the highest probability of success, if TSA personnel never screen luggage or patrol the perimeter, the adversary can choose an attack path that avoids passenger screening such as utilizing a suitcase bomb or an attack from the perimeter.

On the defender side, dedicating more resources to security activities in an area increases the security afforded to that area. However, even with more resources, the TSA wants to avoid being predictable since attackers can exploit this predictability; avoiding the security activities they know will be in place. Thus, I needed to represent threats in a way that accounts for the attacker's ability to observe security in advance and avoid specific security activities, but still represents the benefit of dedicating more resources.

A naïve approach is to represent only a single threat per area and decrease the likelihood of success for that threat as more security activities are put in place. This captures the increase in security for additional security activities, however, it does not account for the attacker's ability to

circumvent security activities. With this method you would simply choose security activities in the order of their relative success making it predictable and exploitable.

The alternative that I chose is to create a list of potential threats that circumvent different combinations of specific security activities. By basing threats on circumventing particular combinations of security activities, I avoid the issue of enumerating all the possible potential threats. Instead the threats are automatically created based on the security activities in an area. However, I also incorporate a cost to the attacker for circumventing more activities to capture the idea of causing maximal damage at minimal cost. Each individual activity has a specific circumvention cost associated with it and more activities circumvented leads to a higher circumvention cost. This cost reflects the additional difficulty of executing an attack against increased security. This difficulty could be due to requiring additional resources, time and other factors for executing an attack. Since attackers can now actively circumvent specific security activities, randomization becomes a key factor in the solutions that are produced because any deterministic strategies can be circumvented.

6.1.2.2 Compact Representation

I introduce a compact representation that exploits similarities in defender security activities to reduce the number of strategies that must be enumerated and considered when finding an optimal solution to SCGs. First, I identify security activities that provide coverage to the same areas, and have the same circumvention costs (i.e., have identical properties). Let $\kappa_i \in K$ represent the sets of security activities that can be grouped together because they have identical properties. Now, instead of reasoning over individual security activities, I reason about groups of identical security

activities $\kappa_i \in K$. A strategy $\sigma_i \in \Sigma_\Theta$ is represented by the number of resources assigned to each set of identical security activities κ_i .

To illustrate this new representation, I provide a concrete example of the full representation versus the compact representation in Tables 6.1 and 6.2. In this example there are 4 security activities and 2 resources. Here, s_1 and s_2 have identical circumvention costs and affect a_1 while s_3 and s_4 have identical circumvention costs and affect a_2 . Table 6.1 presents the full representation with corresponding payoffs and Table 6.2 represents the compact form of the same where κ_1 represents the group s_1 and s_2 and κ_2 represents the group s_3 and s_4 . In both tables, each row represents a single pure strategy for the defender and each column the same for the attacker. Notice in Table 6.1 each strategy $\sigma_i \in \Sigma_\Theta$ is represented by the exact security activities being executed while in Table 6.2 it is only which set $\kappa_i \in K$ each resource has been allocated to.

The key to the compact representation is that each of the security activities from a set $\kappa_i \in K$ will have the same effect on the payoffs. Therefore, it is optimal for the defender to distribute probability uniformly at random across all security activities within a set κ_i , so that all security activities are chosen with equal probability in the solution. Given that the defender strategy uniformly distributes resources among all security activities $s_j \in \kappa_i$, it does not matter which specific security activities the attacker chooses to circumvent from the set κ_i . For any given number of security activities circumvented, the expected payoff to the attacker is identical regardless of which specific activities within the set are chosen. This is because security activities are being selected uniformly at random within the set κ_i . Therefore, I can use a similar compact representation for the attacker strategy space as for the defender, reasoning only over the aggregate number of security activities of each type rather than specific security activities.

Given this, I only need to know how many security activities are selected from each set in order to compute the expected payoffs for each player in the compact representation. For example, examining the second row and second column of Table 6.2 we see that the reward to the defender is -2 and the reward to the attacker is 0. In this case, the defender strategy is to assign 1 resource to activities in κ_1 and 1 resource to activities in κ_2 . Given that she is uniformly distributing these resources, it follows that she will execute s_1 half of the time and s_2 the other half. On the attacker side, we know that the attacker is circumventing one security activity from the set κ_1 . If he circumvents either s_1 or s_2 he will only succeed half of the time. Thus, half of the time the defender receives 4 and the other half -8 for an expectation of -2 (i.e., $4 * .5 + (-8) * .5$). I compute the attacker's reward in the same manner.

	$a_1 : \emptyset$	$a_1 : s_1$	$a_1 : s_2$	$a_2 : \emptyset$	$a_2 : s_3$	$a_2 : s_4$
s_1, s_2	2, -1	4, -3	4, -3	-20, 10	-17, 7	-17, 7
s_1, s_3	2, -1	-8, 3	4, -3	5, -5	-17, 7	8, -8
s_1, s_4	2, -1	-8, 3	4, -3	5, -5	8, -8	-17, 7
s_2, s_3	2, -1	4, -3	-8, 3	5, -5	-17, 7	8, -8
s_2, s_4	2, -1	4, -3	-8, 3	5, -5	8, -8	-17, 7
s_3, s_4	-10, 5	-8, 3	-8, 3	5, -5	8, -8	8, -8

Table 6.1: Example payoffs for sample game.

	$a_1 : \emptyset$	$a_1 : \kappa_1$	$a_2 : \emptyset$	$a_2 : \kappa_2$
κ_1, κ_1	2, -1	4, -3	-20, 10	-17, 7
κ_1, κ_2	2, -1	-2, 0	5, -5	-4.5, -5
κ_2, κ_2	-10, 5	-8, 3	5, -5	8, -8

Table 6.2: Example compact version of sample game.

Given this compact representation for both the defender and attacker, I can compute an optimal mixed strategy of assigning resources over K . Once I have this mixed strategy, I will need to determine an actual strategy for TSA personnel to execute by sampling one of the possible strategies from the mixed strategy I have determined for my compact representation (e.g., one

sample may be $\kappa_2\kappa_2$). Once sampled, I will know exactly how many resources are available to each set $\kappa_i \in K$. Given this resource assignment, I can then sample security activities by selecting m uniformly at random where m is the number of resources assigned to $\kappa_i \in K$. This specific set of security activities for each area under the current resource assignment is a full strategy for TSA personnel to execute.

6.2 Evaluation

When evaluating a game-theoretic model like SCGs there are two important issues that are raised. The first issue is evaluating the value of the security policies generated against alternative approaches. The second issue is with scalability and run-times. To be useful in practice, the approach needs to be able to solve real world challenges. In the following sections I present each of these evaluations.

6.2.1 Security Policy Analysis

For this analysis I examined the security policies generated by my game representation against two other possible solution strategies. The first strategy is a solution concept where resources are distributed uniformly among areas (uniformly random), an approach sometimes used in lieu of a game-theoretic approach. The second strategy uses my new representation, however, it does not allow attackers to circumvent security activities (SCGs without circumvention). That is, I allow the attacker only a single attack strategy per area and simply reduce the value of that strategy as the number of security activities increases. This is a simplified model of an attacker as mentioned

in Section 6.1.2.1. Finally, I included my new representation and allow an intelligent attacker to circumvent specific security activities when planning his mode of attack (SCGs).

I generated 20 random game instances with 10 areas and 3 security activities per area. In each game instance the payoff value of each area for both the defender and attacker are randomly selected from 1 to 50 and the circumvention costs are similarly selected from 1 to 5. I then calculated the optimal solution under the current solution strategy (i.e., uniformly random, SCGs without circumvention, and SCGs). After finding the optimal solution, I determined the expected value for each solution given the assumptions made in SCGs (i.e., attackers are allowed to circumvent specific security activities when planning their attack). For each game instance, I computed the optimal solution varying the number of resources available from 1 to 10 as seen on the x-axis of Figure 6.1. On the y-axis, I present the average expected value obtained by each solution strategy across all 20 game instances. In Figure 6.1 the results show that the uniform policy is outperformed by both game-theoretic approaches with the approach accounting for circumvention strategies performing the best. In fact, an approach that accounts for circumvention strategies is the only one that was able to obtain a positive reward for the defender in the 20 randomly generated game instances and in the 10 resource case obtains a 200% improvement in reward over any other strategy. This shows the benefits of reasoning about an intelligent attacker who will research and exploit deterministic security activities.

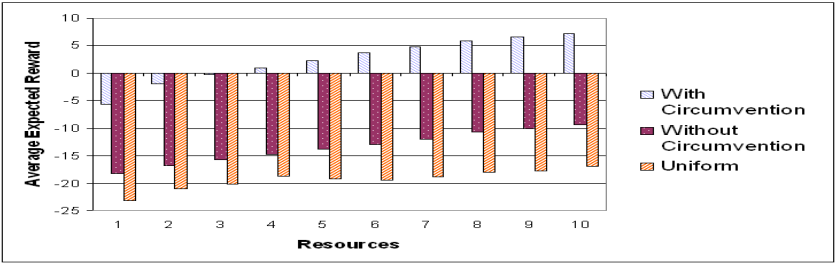


Figure 6.1: Policy Analysis: Increasing resources for 10 areas with 3 security activities per area

6.2.2 Runtime Analysis

I present simulation results focusing on the computational efficiency of my compact method versus the full representation. All experiments are run on a system with an Intel 2 GHz processor and 1 GB of RAM. I used a publicly available linear programming package called GLPK to solve optimization problems as specified in the original DOBSS procedure. For the compact version I use a slightly altered version of DOBSS that is designed specifically for efficiency in the compact representation. The solver was allowed to use up to 700 MB of memory during the solution process. For larger game instances, solving the problem with the full representation runs out of memory and solutions cannot be found. In the results presented below I exclude results for cases where the full representation was not able to produce a result using the allotted memory. I also note that in all experiments both the solution found by the full representation and the solution found by the compact representation are optimal.

To test the solution methods I generated random game instances by randomly selecting payoff values from 1 to 50 and circumvention costs from 1 to 5 for each area. For each experiment I generated 20 random game instances and averaged the results (there is little variance in the run-times for different problem instances). I considered three different scenarios. The first scenario presents results for the case where there is an increasing number of areas, and each area has exactly 3 security activities associated with it. There are 5 resources available for the defender, and each security activity has identical properties (i.e., no security activity has a higher cost for circumvention or higher probability of success) for the area it is associated with. Given the $|A|$ possible areas, for the full representation there are $\binom{3 \cdot |A|}{5}$ possible defender pure strategies and $8 \cdot |A|$ possible attacker pure strategies. Thus, in the 10 area case there are 142,506 defender pure

strategies and 80 attacker pure strategies. Examining Figure 6.2 (a), I show the improvement in runtime of my compact representation over the full representation. For more than 4 areas, the full representation failed to achieve a solution within the memory bounds. For 4 areas, the compact representation runs much faster than the full representation, with a runtime of less than 1 second versus the 177 seconds required by the full representation. In fact, for 10 areas, the compact representation has an average runtime of approximately 1 second, which is still much faster than the full representation for only 4 areas. Even if the number of security activities associated with each area is a relatively small constant my compact representation provides substantial benefits. As the number of similar security activities associated with an area increases, this advantage grows.

In the second scenario, I considered a situation where security activities are distributed randomly across possible areas. The total number of security activities is set similarly to the previous experiment, in that the total number of security activities is three times the number of areas. However, I randomly assigned security activities to areas (with each area having at least one security activity) so the number is no longer uniform across areas. Once again the defender has 5 resources available and security activities have identical properties within an area. It follows that in the full representation, the number of defender pure strategies and attacker pure strategies are identical to the previous scenario. However, the number of strategies in the compact representation for both the defender and attacker may vary. Looking at Figure 6.2 (b), the results show similar benefits for the compact representation in this case as in the previous experiment with a uniform distribution of activities.

In the final scenario, I considered a situation in which there are 10 areas to protect, each area has 3 identical security activities, and I increased the number of resources available to distribute

between these areas. Thus, in the full representation, assuming there are $|R|$ resources available, the defender has $\binom{30}{|R|}$ possible pure strategies and the attacker has 80 possible pure strategies. In Figure 6.3, I increase the number of resources available along the x-axis and show the time to compute a solution in seconds on the y-axis. The full representation is unable to compute a solution for more than 4 resources under these conditions within the allotted memory. On the other hand, the compact representation is able to arrive at a solution for 10 available resources in less than 30 seconds. These results show the benefits of my compact representation in terms of efficiency. Even so, this last result shows that even the compact representation can face runtime difficulties for larger numbers of resources. Given the scale and complexity of airports, finding improved computational methods for this compact representation remains an area for future work. In general though, SCGs provide an efficient framework for addressing more complex security scenarios than traditional security games.

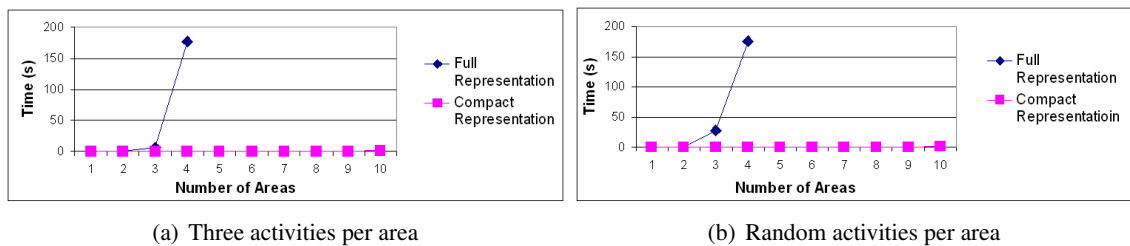


Figure 6.2: X-axis: Areas, Y-axis: Runtime

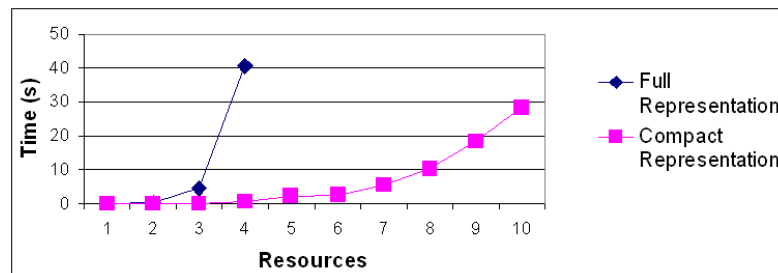


Figure 6.3: Runtime: Increasing resources for 10 areas with 3 security activities per area

Chapter 7: Conclusions

7.1 Summary

Stackelberg games have become crucial in many multiagent applications, and particularly for security applications [Brown et al., 2006; Paruchuri et al., 2008; Jain et al., 2010; Shieh et al., 2012; Pita et al., 2011]; for instance, these games are applied for security scheduling at the Los Angeles International Airport, the Federal Air Marshal Service, and the United States Coast Guard [Jain et al., 2010; Shieh et al., 2012]. In such applications, automated Stackelberg solvers may create an optimal leader strategy. Unfortunately, the standard game-theoretic solution techniques makes three critical assumptions that may not apply in the real-world against a human adversary:

- *Rationality Assumption:* One of the strict assumptions made by standard game-theoretic approaches based on Stackelberg games is that the adversary is perfectly rational. That is, the adversary maximizes his expected value based on the information available.
- *Observability Assumption:* In standard Stackelberg game models it is assumed that the adversary perfectly observes the security force's strategy.

- *Action Space Assumption:* Game-theoretic models require defined potential actions on behalf of all the players involved in order to compute an optimal mixed strategy for the security force.

If these assumptions fail to hold against a human adversary, then it may lead to a severely under-performing strategy when the human adversary deviates from the optimal strategy. In fact, human decisions are guided by their bounded rationality [Simon, 1956, 1969] as opposed to expected-value-maximizing rationality, which may cause them to deviate from their expected optimal strategy. Furthermore, in real-world settings, humans may have limited observability of the security personnel's strategy, giving them a false impression of that strategy. Given this limited information, humans can be biased in their decision making, even deviating from the optimal response of the observed strategy Fox and Rottenstreich [2003]; See et al. [2006a]. Beyond issues of rationality and observability, real-world security scenarios are complex in nature with a large potential action space. Additionally, human adversaries are adaptable and creative allowing them to constantly evolve new attack strategies. Simply enumerating such a large action space may not be practical and, given current solution methods, may not be solvable under reasonable memory or time limitations.

This thesis helped address these three critical assumptions and provides three key contributions:

- **COBRA:** COBRA includes two new key ideas for addressing human adversaries: (i) human anchoring biases drawn from support theory, and (ii) robust approaches for MILPs to address human imprecision. To the best of my knowledge, the effectiveness of each of these key ideas against human adversaries had not been explored in the context of Stackelberg

games. Furthermore, it was unclear how effective the combination of these ideas, being brought together from different fields, would be against humans.

- **MATCH:** MATCH extends the robustness approach of COBRA allowing for a more gradual defense against potential deviations as opposed to a hard cut-off. This new type of graduated robust optimization avoids the complex task of modeling human decision making while still accounting for different degrees of potential deviations.
- **Security Circumvention Games:** SCGs are a novel game-theoretic model, which help address the vast and continually evolving action space of real-world security domains. In creating this model I provide the following contributions: (i) the ability for defenders to guard areas with more than one type of security activity (heterogeneous activities), and (ii) the ability for attackers to choose threats designed to mitigate specific security activities. Furthermore, I designed an efficient solution technique for reasoning over SCGs where I rely on creating a compact representation of each game instance and solving it using a general purpose Stackelberg solver.

The experimental evaluations of COBRA and MATCH validate the usefulness of these approaches against human adversaries compared to both game-theoretically optimal algorithms and an approach known as BRQR, which has previously been shown to be the best performing against human adversaries under perfect observation [Yang et al., 2011]. For COBRA, these results are based on 4 reward structures, in 4 different observability conditions, involving 218 human subjects playing 2960 games in total. For MATCH, these results are based on 104 reward structures, under perfect observation, involving 363 human subjects playing 8823 games in total. Furthermore, MATCH provides significant runtime benefits over BRQR since it exploits the unique

structure of security games [Korzhyk et al., 2011]. The empirical evaluation of SCGs reveals both the benefits to solution quality against intelligent human adversaries and the runtime benefits of such an approach when addressing the vast action space of human adversaries, which may not be practical to enumerate in the standard way. These new approaches for addressing human adversaries provide a significant contribution in transitioning game-theoretic approaches to real-world security domains where security forces will face human adversaries.

7.2 Future Work

In the future the application of game-theoretic approaches will continue to expand to new security domains. For example, security agencies may use game-theoretic approaches to create randomized patrols to protect forests from illegal deforestation or to create randomized patrols in national parks to protect animals from illegal poachers. Each new domain will bring with it new challenges and new complexities. However, the common element in each new security domain will be the human adversaries that security forces will likely face.

This thesis has shown the value of addressing the standard critical assumptions of game-theoretic approaches. However, in each new domain the type of adversary may change, creating a need for more evolved approaches for addressing human adversaries. In addressing the challenges of human adversaries I envision building on the basic insights developed in my thesis, which includes the value of robustly guarding against potential deviations and intelligently representing the underlying action space of security games.

In the short run, the robustness approaches presented could be enhanced by exploiting additional human biases and further developing the graduated robustness ideas of MATCH. For

example, MATCH could be extended to incorporate COBRA's ϵ -robustness for minor deviations, address human anchoring biases under limited observation and to vary the degree of β -robustness based on the magnitude of deviations. That is, instead of being equivalently robust to large deviations as small deviations, an approach could be created that allows the defender to sacrifice more as the magnitude of the deviation increases.

In the longer run, it will be crucial to address the different adversary types that exist in different security domains. Depending on the type of adversary faced, the human-decision-making process may vary. In the future it will be important to explore different psychological profiles and how they affect human-decision-making processes. By examining these crucial psychological profiles it may be possible to create superior algorithms designed specifically for certain criminal types and security domains. Additionally, in the future it will be important to explore the potential multi-attribute aspects of these security problems. MATCH is an algorithm that already indirectly explores the multi-attribute aspects of security problems by taking into consideration the impact a deviation has to the defender and not just the attacker. By doing so, MATCH indirectly accounts for the possibility that the attacker may place some value on the impact his deviation has to the defender. Such an attacker may be willing to sacrifice some of his expected value in order to significantly impact the defender's expected value.

In the future it would be beneficial to create a suite of potential algorithms that exploit different human-decision-making processes for specific psychological profiles, robustly guard against the likely deviations human adversaries will make, and potentially consider multiple attributes within security settings. Additionally, creating better mechanisms for testing and validating new approaches will become critically important. As these approaches continue to transition into real-world domains, appropriately testing and validating the models will become a key concern.

For instance, this may include evaluating things such as robustness to different types of uncertainty (e.g., observational uncertainty, payoff uncertainty, or capability uncertainty), scalability, and deterrence value. For now, no formal methods exist for making comprehensive and reliable evaluations.

Bibliography

- M. Abrahms. What terrorists really want: Terrorist motives and counterterrorism strategy. *International Security*, 32(4):78–105, 2008.
- M. Aghassi and D. Bertsimas. Robust game theory. *Math. Program.*, 107(1):231–273, 2006.
- N. Agmon, G.A. Kaminka, and S. Kraus. Multi-robot adversarial patrolling: Facing a full-knowledge opponent. *Journal of Artificial Intelligence Research*, 42:887–916, 2011.
- G. Allison and P. Zelikow. *Essence of Decision: Explaining the Cuban Missile Crisis*. Longman, 1999.
- B. An, D. Kempe, C. Kiekintveld, E. Shieh, S. Singh, M. Tambe, and Y. Vorobeychik. Security games with limited surveillance. In *AAAI*, pages 1241–1248, 2012.
- C. Archibald and Y. Shoham. Modeling billiards games. In *AAMAS*, pages 193–199, 2009.
- A. Azaria, Z. Rabinovich, S. Kraus, C. V. Goldman, and Y. Gal. Strategic advice provision in repeated human-agent interactions. In *AAAI*, pages 1522–1528, 2012.
- T. Basar and G. J. Olsder. *Dynamic Noncooperative Game Theory*. Academic Press, 1995.
- N. Basilico, N. Gatti, and F. Amigoni. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *AAMAS*, pages 500–503, 2009.
- M. Breton, A. Alj, and A. Haurie. Sequential stackelberg equilibria in two-person games. *Optimization Theory and Applications*, 59(1):71–94, 1988.
- G. Brown, M. Carlyle, J. Kline, and K. Wood. A two-sided optimization for theater ballistic missile defense. *Operations Research*, 53:263–275, 2005.
- G. Brown, M. Carlyle, J. Salmerón, and K. Wood. Defending critical infrastructure. *Interfaces*, 36(6):530–544, 2006.
- E. Brunner, U. Munzel, and M. L. Puri. Rank-score tests in factorial designs with repeated measures. *Journal of Multivariate Analysis*, 70(2):286–317, 1999.
- C. Camerer. *Behavioral Game Theory: Experiments in Strategic Interaction*. Princeton Press, 2003.
- V. Conitzer and T. Sandholm. Computing the optimal strategy to commit to. In *ACM EC*, pages 82–90, 2006.

- C. de Melo, P. Carnevale, and J. Gratch. The effect of expression of anger and happiness in computer agents on negotiations with humans. In *AAMAS*, pages 937–944, 2011.
- DHS. United States Department of Homeland Security: Border Security. September 2012a. URL <http://www.dhs.gov/border-security/>.
- DHS. United States Department of Homeland Security: Transportation Security. September 2012b. URL <http://www.dhs.gov/topic/transportation-security/>.
- EPA. US Environmental Protection Agency. September 2012. URL <http://www.epa.gov/>.
- C. R. Fox and R. T. Clemen. Subjective probability assessment in decision analysis: Partition dependence and bias toward the ignorance prior. *Management Science*, 51(9):1417–1432, 2005.
- C. R. Fox and Y. Rottenstreich. Partition priming in judgement under uncertainty. *Psychological Science*, 14:195–200, 2003.
- M. Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(100):675–701, 1937.
- N. Gatti. Game theoretical insights in strategic patrolling: Model and algorithm in normal-form. In *ECAI*, pages 403–407, 2008.
- P. Gill and J. K. Young. Comparing role-specific terrorist profiles. 2011. URL <http://dx.doi.org/10.2139/ssrn.1782008>.
- P. E. Greenwood and M. S. Nikulin. *A Guide to Chi-squared Testing*. John Wiley & Sons, Inc., 1996.
- GTI. Global tiger initiative official website. September 2012. URL <http://www.globaltigerinitiative.org/>.
- G. Haim, Y. Gal, and S. Kraus. A cultural sensitive agent for human-computer negotiation. In *AAMAS*, pages 451–458, 2012.
- J. C. Harsanyi and R. Selten. A generalized Nash solution for two-person bargaining games with incomplete information. *Management Science*, 18(5):80–106, 1972.
- M. Jain, J. Tsai, J. Pita, C. Kiekintveld, S. Rathi, M. Tambe, and F. Ordóñez. Software assistants for randomized patrol planning for the LAX Airport Police and the Federal Air Marshals Service. *Interfaces*, 40:267–290, 2010.
- M. Jain, D. Korzhyk, O. Vanek, V. Conitzer, M. Pechoucek, and M. Tambe. A double oracle algorithm for zero-sum security games on graphs. In *AAMAS*, pages 327–334, 2011a.
- M. Jain, M. Tambe, and C. Kiekintveld. Quality-bounded solutions for finite Bayesian Stackelberg games: Scaling up. In *AAMAS*, pages 997–1004, 2011b.
- M. P. Johnson, F. Fang, M. Tambe, and H. J. Albers. Patrol strategies to maximize pristine forest area. In *AAAI*, pages 37–41, 2012.

- D. Kahneman and A. Tversky. Subjective probability: A judgement of representativeness. *Cognitive Psychology*, 3:430–454, 1972.
- D. Kahneman and A. Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–292, 1979.
- G. L. Keeney and D. von Winterfeldt. Identifying and structuring the objectives of terrorists. *Risk Analysis*, 30(12):1803–1816, 2010.
- C. Kiekintveld, M. Jain, J. Tsai, J. Pita, M. Tambe, and F. Ordóñez. Computing optimal randomized resource allocations for massive security games. In *AAMAS*, pages 689–696, 2009.
- C. Kiekintveld, J. Marecki, and M. Tambe. Approximate method for infinite bayesian stackelberg games: Modeling distributional payoff uncertainty. In *AAMAS*, pages 1005–1012, 2011.
- D. J. Koehler and G. James. Probability matching in choice under uncertainty: Intuition versus deliberation. *Cognition*, 113:123–127, 2009.
- D. Korzhyk, Z. Yin, C. Kiekintveld, V. Conitzer, and M. Tambe. Stackelberg vs. Nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness. volume 41, pages 297–327, 2011.
- LAPD. Official website of THE LOS ANGELES POLICE DEPARTMENT. September 2012. URL <http://www.lapdonline.org/>.
- LAWA. LAX - general description. September 2012. URL http://www.lawa.org/welcome_LAX.aspx?id=40.
- G. Leitmann. On generalized Stackelberg strategies. *Optimization Theory and Applications*, 26(4):637–643, 1978.
- K. Lye and J. M. Wing. Game strategies in network security. *International Journal of Information Security*, 4(1–2):71–86, 2005.
- S. Marsella, J. Gratch, and P. Petta. Computational models of emotion. In *A blueprint for affective computing: A sourcebook and manual*. Oxford: Oxford University Press, 2010.
- M. D. McCubbins, M. B. Turner, and N. Weller. The theory of minds within the theory of games. In *International Conference on Artificial Intelligence*, 2012.
- R. McKelvey and T. Palfrey. Quantal response equilibria for normal form games. *Games and Economic Behavior*, 10(1):6–38, 1995.
- A. R. Morral and B. A. Jackson. Understanding the role of deterrence in counterterrorism security. *RAND Corporation, OP-281-RC*, 2009. URL http://www.rand.org/pubs/occasional_papers/OP281/.
- NATCA. Air traffic controllers. September 2012. URL http://www.natca.org/who_we_are.aspx?zone=Who We Are&pID=254#p254.

- A. Nilim and L. E. Ghaoui. Robust control in markov decision problems with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.
- Nobelprize.org. The sveriges riksbank prize in economic sciences in memory of alfred nobel 2000. September 2012. URL http://www.nobelprize.org/nobel_prizes/economics/laureates/2000/.
- E. Nudelman, J. Wortman, Y. Shoham, and K. Leyton-Brown. Run the GAMUT: A comprehensive approach to evaluating game-theoretic algorithms. In *AAMAS*, pages 880–887, 2004.
- F. Ordóñez and N. Stier-Moses. Robust wardrop equilibrium. In *NET-COOP*, volume 4465, pages 247–256, 2007.
- P. Paruchuri, J. P. Pearce, J. Marecki, M. Tambe, F. Ordóñez, and S. Kraus. Playing games with security: An efficient exact algorithm for Bayesian Stackelberg games. In *AAMAS*, pages 895–902, 2008.
- J. Pita, M. Jain, C. Western, C. Portway, M. Tambe, F. Ordóñez, S. Kraus, and P. Paruchuri. Deployed ARMOR protection: The application of a game theoretic model for security at the Los Angeles International Airport. In *AAMAS*, pages 125–132, 2008.
- J. Pita, M. Jain, M. Tambe, F. Ordóñez, and S. Kraus. Robust solutions to stackelberg games: Addressing bounded rationality and limited observations in human cognition. *Artificial Intelligence Journal*, 174(15):1142–1171, 2010.
- J. Pita, C. Kiekintveld, M. Tambe, E. Steigerwald, and S. Cullen. GUARDS - game theoretic security allocation on a national scale. In *AAMAS*, pages 37–44, 2011.
- M. R. Pogrebin. *About Criminals: A View of the Offenders' World*. SAGE, 2012.
- R. Porter, A. Ronen, Y. Shoham, and M. Tennenholtz. Mechanism design with execution uncertainty. In *UAI*, pages 414–421, 2002.
- L. Richardson. *What Terrorist Want: Understanding the Enemy, Containing the Threat*. Random House, 2006.
- H. Rosoff and R. John. Decision analysis by proxy for the rational terrorist. In *QRASA*, pages 25–32, 2009.
- A. Rubinstein. *Modeling Bounded Rationality*. MIT Press, 1998.
- T. Sandler and D. G. Arce M. Terrorism and game theory. *Simulation and Gaming*, 34(3): 319–337, 2003.
- K. E. See, C. R. Fox, and Y. Rottenstreich. Between ignorance and truth: Partition dependence and learning in judgment under uncertainty. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 32:1385–1402, 2006a.
- K. E. See, C. R. Fox, and Y. Rottenstreich. Between ignorance and truth: Partition dependence and learning in judgment under uncertainty. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 32:1385–1402, 2006b.

- R. Selten. Evolutionary stability in extensive two-person games - correction and further development. *Mathematical Social Sciences*, 16:223–266, 1988.
- E. Shieh, B. An, R. Yang, M. Tambe, C. Baldwin, J. DiRenzo, B. Maule, and G. Meyer. PROTECT: An application of computational game theory for the security of the ports of the United States. In *AAAI*, pages 2173–2179, 2012.
- H. Simon. Rational choice and the structure of the environment. *Psychological Review*, 63(2): 129–138, 1956.
- H. Simon. *Sciences of the Artificial*. MIT Press, 1969.
- C. Starmer. Developments in non-expected utility theory: The hunt for a descriptive theory of choice under risk. *Journal of Economic Literature*, XXXVIII:332–382, 2000.
- D. Stevens, T. Hamilton, M. Schaffer, D. Dunham-Scott, J. J. Medby, E. W. Chan, J. Gibson, M. Eisman, R. Mesic, C. T. Kelly Jr, J. Kim, T. LaTourrette, and J. Riley. Implementing security improvement options at Los Angeles International Airport. *RAND Corporation*, 2006. URL http://www.rand.org/pubs/documented_briefings/2006/RAND_DB499-1.pdf.
- S. Tijs. Nash equilibria for noncooperative n-person games in normal form. *SIAM Review*, 23(2): 225–237, 1981.
- TSA. Transportation Security Administration — U.S. Department of Homeland Security. September 2012. URL <http://www.tsa.gov/>.
- J. Tsai, Z. Yin, J. Kwak, D. Kempe, C. Kiekintveld, and M. Tambe. Urban security: Game-theoretic resource allocation in networked physical domains. In *AAAI*, pages 881–886, 2010.
- A. Tversky and D. Koehler. Support theory: A nonextensional representation of subjective probability. *Psychological Review*, 101(4):547–567, 1994.
- J. von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100:295–320, 1927.
- H. von Stackelberg. *Marktform und Gleichgewicht*. Springer, 1934.
- B. von Stengel and S. Zamir. Leadership with commitment to mixed strategies. In *CDAM Research Report LSE-CDAM-2004-01, London School of Economics*, 2004.
- R. Wilcox. How many discoveries have been lost by ignoring modern statistical methods? *American Psychologist*, 53(3):300–314, 1998.
- R. Wilcox. *Applying Contemporary Statistical Techniques*. Academic Press, 2003.
- R. Wilcox. *Introduction to Robust Estimation and Hypothesis Testing*. Academic Press, 2005.
- J. R. Wright and K. Leyton-Brown. Beyond equilibrium: Predicting human behavior in normal-form games. In *AAAI*, pages 901–907, 2010.
- R. Yang, C. Kiekintveld, F. Ordóñez, M. Tambe, and R. John. Improving resource allocation strategy against human adversaries in security games. In *IJCAI*, pages 458–464, 2011.

- Z. Yin and M. Tambe. A unified method for handling discrete and continuous uncertainty in Bayesian Stackelberg games. In *AAMAS*, pages 855–862, 2012.
- Z. Yin, M. Jain, M. Tambe, and F. Ordóñez. Risk-averse strategies for security games with execution and observational uncertainty. In *AAAI*, pages 758–763, 2012a.
- Z. Yin, A. Jiang, M. Johnson, M. Tambe, C. Kiekintveld, K. Leyton-Brown, T. Sandholm, and J. Sullivan. TRUSTS: Scheduling randomized patrols for fare inspection in transit systems. In *IAAI*, 2012b.
- K. K. Yuen. The two-sample trimmed t for unequal population variances. *Biometrika*, 61:165–170, 1974.
- J. Zhuang and V. Bier. Secrecy and deception at equilibrium, with applications to anti-terrorism resource allocation. *Defence and Peace Economics*, 22(1):43–61, 2011.

Appendix A: Statistical Significance Tests

As noted earlier, I was unable to use classical tests such as the T-test to judge statistical significance and hence different types of tests were run on my data. Additionally, because the experimental setup for COBRA was different than that used for MATCH, I use separate statistical tests for the COBRA data and the MATCH data. I will first review the statistical tests used for COBRA and why I chose these tests and then I will do the same for MATCH.

A.1 Statistical Significance Tests for COBRA

Among the reward structures used to test COBRA, reward structure four was treated separately as it is a zero-sum game. I explain the tests I used and the reasons these were chosen in the following. Reward structure one and reward structure two are structures with higher penalties to the leader when the follower deviates from strong Stackelberg equilibrium (SSE) assumptions. Given this similarity in reward structures one and two and the similarity of the results they produced I first ran a two-way Friedman test for repeated observations Friedman [1937] in the unobserved observation condition and a two-way Friedman test (not for repeated observations) in the 5, 20, and unlimited observation conditions between them. A two-way test is a test that examines two variables within an experiment. In my case the two variables are reward structure and algorithm. I ran these tests to be certain that reward structure had an impact on the results produced for reward structures one and two. If reward structure did not have an impact on the results produced the data sets can be combined into one large data set.

In the unobserved observation condition, the conclusion based on this test was that the reward structures did have an impact on the results, however, in the 5, 20, and unlimited observation conditions I found that they did not. Since reward structure did not influence the results in these cases, the data sets from reward structures one and two were combined into one large data set for the 5, 20, and unlimited observation cases. This arrangement left reward structure three separated by itself (since reward structure four was kept separate as a zero-sum game), but when combining reward structure three with one and two, I find that it had an impact on performance and hence separation was justified. The null hypothesis in all of my tests is that the results produced between any two algorithms are actually identical. This hypothesis is rejected with a p-value of .05 or less. Given this setup I had the following statistical tests.

First, the unobserved case was treated separately from other cases. In the unobserved case, subjects were only given the reward structure and were asked to make a choice. Consequently, the choice they made would be made irrespective of the actual strategy employed. Therefore I was able to take the choice made by an individual subject and employ it across DOBSS, COBRA(0, ϵ),

COBRA(α, ϵ), COBRA(C, ϵ), UNIFORM, and MAXIMIN for a particular reward structure. This means that for a particular reward structure in the unobserved case, the door choice made by a single subject was used against all strategies. However, notice that a choice made in a single reward structure was not used for the other three reward structures. For the 5, 20, and unlimited observation cases it was necessary to obtain 40 sample points (subject door choices) for each algorithm in each reward structure. This leads to the following statistical significance tests:

- *Test run for the unobserved case:* I ran the Brunner-Puri test for repeated observations Brunner et al. [1999] in this case. I chose to run the Brunner-Puri test because the Brunner-Puri method is better suited to non-continuous distributions with discrete data points for one-way designs. It is also necessary to use a test that deals with repeated observations since the choices made by subjects were repeated across algorithms. An alternative and more well known method to the Brunner-Puri method is the Friedman test for repeated measures, however, the Brunner-Puri method is a more robust method.
- *Test run for the 5, 20, and unlimited observation cases:* For these cases I tested each of the algorithms separately (i.e. I did not use one subject's choice across all algorithms as in the unobserved case) so it is necessary to use a different type of test. Once again due to the non-normal distribution of my data I chose a more robust method for this test. In particular I chose to run Yuen's test for comparing trimmed means Yuen [1974]. For my tests I used a standard 20% trimmed mean. A trimmed mean refers to a situation where a certain proportion of the largest and smallest sample points are removed and the remaining sample points are averaged. This is typically done to help reduce variance in data collections that may have extreme outliers that can skew data sets Wilcox [1998, 2005].

A.2 Statistical Significance Tests for MATCH

In the experimental setup used in MATCH subjects were required to play all strategies for a given reward structure. This led to a within-subjects experimental design which allowed me to directly compare the results of each strategy for each subject in each reward structure. Given that a direct comparison could be made, this allowed me to evaluate the statistical significance of my results using the bootstrap-t method [Wilcox, 2003]. The bootstrap-t method is a more robust statistical method than the classic t-test, which is necessary given the non-normal discrete distribution of my results. The bootstrap-t method also allows for direct comparison between two strategies to determine if one strategy is statistically significantly better than an alternate strategy. Thus, the use of the bootstrap-t method over the classic t-test in these experiments is justified.

Appendix B: Reward Structures

Doors	0	1	2	3	4	5	6	7
Subject Reward	1	9	5	6	7	1	10	3
Subject Penalty	-2	-4	-3	-3	-3	-2	-4	-3
Defender Reward	1	4	2	3	4	1	5	2
Defender Penalty	-5	-8	-1	-6	-5	-1	-7	-7

Table B.1: Reward structure 1

Doors	0	1	2	3	4	5	6	7
Subject Reward	8	5	3	10	1	3	9	4
Subject Penalty	-3	-2	-3	-2	-3	-3	-2	-3
Defender Reward	4	3	1	5	1	2	5	2
Defender Penalty	-8	-10	-1	-8	-1	-3	-11	-5

Table B.2: Reward structure 2

Doors	0	1	2	3	4	5	6	7
Subject Reward	8	5	2	10	1	3	9	4
Subject Penalty	-3	-3	-3	-3	-3	-3	-3	-3
Defender Reward	4	3	1	5	1	2	5	2
Defender Penalty	-8	-5	-1	-10	-5	-3	-9	-6

Table B.3: Reward structure 3

Doors	0	1	2	3	4	5	6	7
Subject Reward	8	5	2	10	1	3	9	4
Subject Penalty	-3	-3	-3	-3	-3	-3	-3	-3
Defender Reward	3	3	3	3	3	3	3	3
Defender Penalty	-8	-5	-2	-10	-1	-3	-9	-4

Table B.4: Reward structure 4

Doors	0	1	2	3	4	5	6	7
Subject Reward	10	8	3	7	6	7	8	2
Subject Penalty	-7	-4	-6	-8	-4	-2	-9	-3
Defender Reward	2	6	7	7	8	8	6	9
Defender Penalty	-8	-10	-3	-1	-10	-5	-2	-5

Table B.5: Reward structure 5

Doors	0	1	2	3	4	5	6	7
Subject Reward	9	8	2	9	10	1	10	1
Subject Penalty	-10	-1	-10	-8	-4	-10	-5	-3
Defender Reward	3	8	9	9	7	7	4	1
Defender Penalty	-10	-2	-5	-1	-7	-6	-2	-1

Table B.6: Reward structure 6

Doors	0	1	2	3	4	5	6	7
Subject Reward	8	6	1	3	1	7	3	5
Subject Penalty	-6	-9	-3	-7	-7	-2	-5	-2
Defender Reward	5	3	8	3	3	4	3	6
Defender Penalty	-2	-5	-4	-6	-3	-10	-7	-2

Table B.7: Reward structure 7

Doors	0	1	2	3	4	5	6	7
Subject Reward	3	7	3	9	2	9	7	8
Subject Penalty	-4	-8	-5	-8	-9	-4	-1	-6
Defender Reward	5	9	10	2	10	4	8	8
Defender Penalty	-10	-4	-9	-3	-10	-10	-2	-5

Table B.8: Reward structure 8

Doors	0	1	2	3	4	5	6	7
Subject Reward	1	8	7	6	7	4	5	7
Subject Penalty	-1	-5	-4	-6	-5	-2	-3	-6
Defender Reward	6	8	7	7	10	10	7	7
Defender Penalty	-6	-7	-5	-6	-8	-10	-3	-4

Table B.9: Reward structure 9

Doors	0	1	2	3	4	5	6	7
Subject Reward	7	8	1	5	5	6	6	8
Subject Penalty	-4	-4	-3	-1	-2	-6	-5	-6
Defender Reward	7	7	1	6	7	4	7	6
Defender Penalty	-5	-6	-5	-3	-7	-1	-8	-7

Table B.10: Reward structure 10

Doors	0	1	2	3	4	5	6	7
Subject Reward	9	3	4	5	3	6	1	7
Subject Penalty	-4	-1	-1	-1	-8	-2	-5	-4
Defender Reward	9	3	8	9	1	8	4	1
Defender Penalty	-5	-4	-5	-1	-10	-6	-4	-9

Table B.11: Reward structure 11

Doors	0	1	2	3	4	5	6	7
Subject Reward	6	8	1	2	7	4	1	2
Subject Penalty	-8	-5	-5	-2	-9	-3	-2	-1
Defender Reward	10	7	4	1	10	1	7	8
Defender Penalty	-6	-5	-3	-1	-8	-5	-3	-2

Table B.12: Reward structure 12

Doors	0	1	2	3	4	5	6	7
Subject Reward	5	5	1	10	5	3	6	5
Subject Penalty	-2	-1	-7	-4	-3	-3	-7	-1
Defender Reward	7	6	2	1	5	2	8	9
Defender Penalty	-5	-5	-10	-6	-4	-4	-10	-9

Table B.13: Reward structure 13

Doors	0	1	2	3	4	5	6	7
Subject Reward	7	6	9	5	1	1	4	1
Subject Penalty	-4	-7	-6	-7	-4	-1	-4	-2
Defender Reward	7	10	5	4	6	10	8	8
Defender Penalty	-9	-5	-8	-7	-4	-10	-4	-8

Table B.14: Reward structure 14

Doors	0	1	2	3	4	5	6	7
Subject Reward	5	7	5	4	4	4	4	10
Subject Penalty	-4	-6	-5	-3	-6	-4	-4	-10
Defender Reward	2	3	7	9	6	1	4	7
Defender Penalty	-5	-6	-5	-6	-6	-1	-5	-7

Table B.15: Reward structure 15

Doors	0	1	2	3	4	5	6	7
Subject Reward	5	10	4	7	7	7	7	6
Subject Penalty	-3	-10	-3	-8	-8	-4	-5	-5
Defender Reward	7	7	6	2	6	10	2	3
Defender Penalty	-3	-6	-3	-2	-10	-7	-1	-4

Table B.16: Reward structure 16

Doors	0	1	2	3	4	5	6	7
Subject Reward	9	4	8	1	9	6	7	6
Subject Penalty	-6	-2	-6	-1	-6	-5	-7	-3
Defender Reward	7	7	10	7	6	1	10	6
Defender Penalty	-4	-3	-6	-8	-5	-1	-7	-1

Table B.17: Reward structure 17

Doors	0	1	2	3	4	5	6	7
Subject Reward	6	7	6	4	7	4	2	7
Subject Penalty	-10	-4	-5	-5	-8	-5	-1	-9
Defender Reward	3	6	3	4	6	6	5	9
Defender Penalty	-10	-4	-1	-6	-4	-5	-4	-9

Table B.18: Reward structure 18

Doors	0	1	2	3	4	5	6	7
Subject Reward	8	1	3	4	5	4	3	4
Subject Penalty	-8	-3	-5	-5	-5	-1	-5	-8
Defender Reward	7	9	9	10	5	7	4	4
Defender Penalty	-4	-3	-10	-8	-7	-4	-4	-7

Table B.19: Reward structure 19

Doors	0	1	2	3	4	5	6	7
Subject Reward	4	6	1	5	6	7	5	6
Subject Penalty	-1	-7	-1	-4	-8	-9	-6	-5
Defender Reward	4	7	1	4	5	5	8	10
Defender Penalty	-8	-9	-6	-1	-8	-6	-10	-6

Table B.20: Reward structure 20

Doors	0	1	2	3	4	5	6	7
Subject Reward	7	9	7	1	4	7	6	5
Subject Penalty	-5	-4	-2	-2	-1	-6	-3	-3
Defender Reward	3	8	6	7	5	4	3	5
Defender Penalty	-5	-10	-8	-10	-5	-1	-6	-5

Table B.21: Reward structure 21

Doors	0	1	2	3	4	5	6	7
Subject Reward	5	3	6	7	7	3	7	8
Subject Penalty	-10	-8	-9	-4	-6	-1	-8	-6
Defender Reward	2	2	5	7	10	6	8	6
Defender Penalty	-2	-7	-5	-10	-9	-5	-10	-6

Table B.22: Reward structure 22

Doors	0	1	2	3	4	5	6	7
Subject Reward	3	1	5	7	9	1	3	10
Subject Penalty	-7	-2	-7	-4	-3	-8	-4	-5
Defender Reward	7	10	7	10	1	2	4	5
Defender Penalty	-5	-6	-7	-3	-5	-8	-7	-5

Table B.23: Reward structure 23

Doors	0	1	2	3	4	5	6	7
Subject Reward	6	8	4	1	7	6	4	8
Subject Penalty	-7	-9	-8	-5	-6	-7	-1	-3
Defender Reward	7	10	3	6	10	4	6	6
Defender Penalty	-5	-5	-6	-3	-10	-7	-3	-4

Table B.24: Reward structure 24

Doors	0	1	2	3	4	5	6	7
Subject Reward	6	10	5	3	1	4	2	4
Subject Penalty	-9	-8	-8	-8	-8	-7	-8	-8
Defender Reward	8	1	8	5	8	2	10	5
Defender Penalty	-6	-5	-3	-4	-5	-5	-8	-1

Table B.25: Reward structure 25

Doors	0	1	2	3	4	5	6	7
Subject Reward	6	7	1	6	6	6	5	5
Subject Penalty	-6	-8	-1	-6	-7	-7	-5	-5
Defender Reward	6	7	8	4	10	2	8	3
Defender Penalty	-5	-5	-8	-5	-10	-4	-9	-6

Table B.26: Reward structure 26

Doors	0	1	2	3	4	5	6	7
Subject Reward	6	1	2	6	1	4	2	3
Subject Penalty	-5	-4	-3	-8	-1	-7	-6	-5
Defender Reward	7	3	7	9	10	6	7	7
Defender Penalty	-10	-4	-8	-9	-10	-7	-7	-10

Table B.27: Reward structure 27

Doors	0	1	2	3	4	5	6	7
Subject Reward	6	3	1	5	2	4	6	6
Subject Penalty	-6	-5	-1	-7	-7	-6	-6	-2
Defender Reward	10	2	9	8	6	7	7	1
Defender Penalty	-10	-1	-8	-9	-5	-6	-5	-1

Table B.28: Reward structure 28

Doors	0	1	2	3	4	5	6	7
Subject Reward	5	5	3	5	4	10	3	5
Subject Penalty	-8	-3	-7	-3	-5	-10	-4	-7
Defender Reward	6	7	4	7	7	8	8	4
Defender Penalty	-5	-6	-4	-4	-7	-6	-10	-1

Table B.29: Reward structure 29

Doors	0	1	2	3	4	5	6	7
Subject Reward	7	2	8	8	7	1	5	8
Subject Penalty	-6	-4	-9	-10	-8	-1	-4	-9
Defender Reward	6	1	7	7	10	4	2	8
Defender Penalty	-5	-1	-7	-9	-9	-5	-3	-8

Table B.30: Reward structure 30

Doors	0	1	2	3	4	5	6	7
Subject Reward	2	6	3	7	5	5	1	5
Subject Penalty	-2	-6	-3	-6	-6	-5	-1	-7
Defender Reward	4	3	7	5	5	1	3	4
Defender Penalty	-4	-5	-7	-8	-6	-1	-5	-3

Table B.31: Reward structure 31

Doors	0	1	2	3	4	5	6	7
Subject Reward	4	4	5	1	5	5	3	10
Subject Penalty	-10	-4	-9	-3	-9	-4	-3	-8
Defender Reward	7	3	8	8	7	5	9	6
Defender Penalty	-5	-5	-4	-1	-9	-3	-10	-5

Table B.32: Reward structure 32

Doors	0	1	2	3	4	5	6	7
Subject Reward	10	4	6	4	5	4	5	7
Subject Penalty	-9	-5	-8	-5	-9	-9	-7	-10
Defender Reward	6	6	1	2	6	3	6	4
Defender Penalty	-8	-5	-3	-1	-7	-7	-8	-8

Table B.33: Reward structure 33

Doors	0	1	2	3	4	5	6	7
Subject Reward	9	4	2	4	3	1	10	10
Subject Penalty	-2	-7	-4	-4	-2	-3	-3	-10
Defender Reward	8	5	8	4	8	2	8	3
Defender Penalty	-9	-10	-3	-4	-8	-4	-8	-1

Table B.34: Reward structure 34

Doors	0	1	2	3	4	5	6	7
Subject Reward	2	6	6	5	5	10	5	5
Subject Penalty	-3	-7	-9	-7	-5	-10	-4	-10
Defender Reward	6	7	3	7	10	3	6	7
Defender Penalty	-5	-10	-3	-4	-5	-7	-6	-5

Table B.35: Reward structure 35

Doors	0	1	2	3	4	5	6	7
Subject Reward	3	10	4	8	5	4	8	4
Subject Penalty	-10	-10	-8	-6	-7	-6	-1	-1
Defender Reward	4	4	5	3	3	3	1	3
Defender Penalty	-9	-7	-7	-6	-8	-10	-7	-1

Table B.36: Reward structure 36

Doors	0	1	2	3	4	5	6	7
Subject Reward	4	10	6	1	9	9	7	1
Subject Penalty	-4	-4	-3	-4	-6	-1	-5	-1
Defender Reward	4	6	6	4	3	9	4	10
Defender Penalty	-3	-5	-5	-10	-5	-7	-7	-2

Table B.37: Reward structure 37

Doors	0	1	2	3	4	5	6	7
Subject Reward	6	2	9	6	3	5	3	3
Subject Penalty	-4	-10	-8	-1	-10	-10	-10	-8
Defender Reward	6	10	5	8	4	6	3	5
Defender Penalty	-7	-4	-10	-5	-2	-6	-1	-3

Table B.38: Reward structure 38

Doors	0	1	2	3	4	5	6	7
Subject Reward	7	6	8	4	10	2	6	5
Subject Penalty	-8	-10	-8	-3	-6	-2	-4	-8
Defender Reward	9	7	4	4	10	2	5	6
Defender Penalty	-10	-3	-7	-3	-5	-5	-4	-3

Table B.39: Reward structure 39

Doors	0	1	2	3	4	5	6	7
Subject Reward	2	10	7	9	1	3	1	4
Subject Penalty	-6	-2	-2	-8	-9	-7	-4	-8
Defender Reward	5	7	7	4	6	1	4	6
Defender Penalty	-6	-3	-1	-7	-7	-7	-7	-7

Table B.40: Reward structure 40

Doors	0	1	2	3	4	5	6	7
Subject Reward	9	6	7	7	5	10	4	7
Subject Penalty	-9	-4	-5	-7	-4	-10	-3	-4
Defender Reward	4	5	6	2	1	3	3	1
Defender Penalty	-6	-6	-9	-3	-1	-5	-5	-2

Table B.41: Reward structure 41

Doors	0	1	2	3	4	5	6	7
Subject Reward	5	9	7	9	7	5	7	9
Subject Penalty	-3	-8	-3	-10	-5	-1	-4	-8
Defender Reward	4	1	8	7	1	9	6	4
Defender Penalty	-5	-2	-8	-8	-1	-9	-6	-7

Table B.42: Reward structure 42

Doors	0	1	2	3	4	5	6	7
Subject Reward	5	5	8	10	4	3	4	4
Subject Penalty	-7	-7	-9	-10	-7	-5	-5	-6
Defender Reward	5	6	5	1	7	6	6	1
Defender Penalty	-5	-7	-7	-1	-7	-8	-7	-3

Table B.43: Reward structure 43

Doors	0	1	2	3	4	5	6	7
Subject Reward	9	8	4	10	6	6	9	7
Subject Penalty	-8	-7	-4	-10	-5	-5	-8	-5
Defender Reward	10	4	3	5	1	3	5	3
Defender Penalty	-10	-8	-9	-8	-3	-4	-8	-8

Table B.44: Reward structure 44

Doors	0	1	2	3	4	5	6	7
Subject Reward	5	6	4	8	2	4	10	7
Subject Penalty	-6	-8	-3	-5	-3	-6	-10	-5
Defender Reward	4	4	4	1	4	7	7	4
Defender Penalty	-5	-5	-2	-3	-1	-6	-7	-3

Table B.45: Reward structure 45

Doors	0	1	2	3	4	5	6	7
Subject Reward	8	7	7	1	7	3	7	8
Subject Penalty	-3	-2	-3	-1	-5	-1	-5	-6
Defender Reward	6	1	2	5	4	5	3	6
Defender Penalty	-6	-1	-3	-8	-7	-6	-4	-7

Table B.46: Reward structure 46

Doors	0	1	2	3	4	5	6	7
Subject Reward	5	1	7	5	2	5	2	1
Subject Penalty	-7	-4	-10	-6	-4	-7	-3	-1
Defender Reward	6	3	6	2	4	5	4	1
Defender Penalty	-7	-4	-6	-1	-6	-6	-5	-2

Table B.47: Reward structure 47

Doors	0	1	2	3	4	5	6	7
Subject Reward	10	4	1	7	8	5	5	5
Subject Penalty	-8	-6	-1	-8	-9	-6	-8	-4
Defender Reward	7	4	1	8	3	8	8	2
Defender Penalty	-8	-5	-4	-7	-4	-7	-9	-1

Table B.48: Reward structure 48

Doors	0	1	2	3	4	5	6	7
Subject Reward	10	6	4	5	7	5	3	8
Subject Penalty	-10	-7	-5	-5	-3	-7	-4	-7
Defender Reward	7	7	6	8	7	3	5	10
Defender Penalty	-6	-4	-8	-9	-6	-6	-4	-10

Table B.49: Reward structure 49

Doors	0	1	2	3	4	5	6	7
Subject Reward	6	1	2	8	9	4	4	5
Subject Penalty	-7	-1	-3	-10	-5	-5	-7	-4
Defender Reward	10	3	6	8	8	1	8	6
Defender Penalty	-4	-2	-2	-7	-7	-1	-4	-5

Table B.50: Reward structure 50

Doors	0	1	2	3	4	5	6	7
Subject Reward	5	4	6	5	5	10	5	4
Subject Penalty	-6	-6	-5	-6	-6	-10	-5	-7
Defender Reward	7	1	4	4	6	4	1	2
Defender Penalty	-5	-3	-7	-4	-5	-2	-1	-3

Table B.51: Reward structure 51

Doors	0	1	2	3	4	5	6	7
Subject Reward	7	4	6	9	5	4	1	4
Subject Penalty	-3	-2	-5	-5	-2	-1	-1	-1
Defender Reward	6	2	2	7	1	4	6	8
Defender Penalty	-4	-1	-3	-7	-2	-6	-6	-5

Table B.52: Reward structure 52

Doors	0	1	2	3	4	5	6	7
Subject Reward	3	3	4	10	4	5	5	7
Subject Penalty	-6	-6	-7	-10	-6	-7	-3	-10
Defender Reward	4	2	4	1	6	6	6	4
Defender Penalty	-6	-1	-6	-1	-7	-7	-7	-6

Table B.53: Reward structure 53

Doors	0	1	2	3	4	5	6	7
Subject Reward	7	9	8	6	5	1	4	6
Subject Penalty	-8	-8	-9	-5	-7	-1	-5	-6
Defender Reward	10	5	9	8	6	1	3	7
Defender Penalty	-8	-7	-8	-4	-6	-4	-2	-7

Table B.54: Reward structure 54

Doors	0	1	2	3	4	5	6	7
Subject Reward	8	8	1	6	9	5	8	5
Subject Penalty	-7	-5	-1	-6	-8	-1	-4	-2
Defender Reward	7	3	6	6	5	7	1	2
Defender Penalty	-6	-9	-1	-7	-3	-2	-4	-4

Table B.55: Reward structure 55

Doors	0	1	2	3	4	5	6	7
Subject Reward	8	4	1	3	3	2	9	5
Subject Penalty	-4	-3	-1	-3	-2	-1	-9	-4
Defender Reward	1	6	6	1	10	6	2	8
Defender Penalty	-4	-10	-6	-3	-10	-7	-3	-7

Table B.56: Reward structure 56

Doors	0	1	2	3	4	5	6	7
Subject Reward	7	5	10	6	8	1	5	9
Subject Penalty	-7	-9	-6	-2	-2	-3	-6	-6
Defender Reward	9	1	7	3	10	7	8	8
Defender Penalty	-7	-5	-7	-4	-7	-5	-5	-6

Table B.57: Reward structure 57

Doors	0	1	2	3	4	5	6	7
Subject Reward	5	9	6	5	9	5	3	6
Subject Penalty	-10	-4	-5	-1	-9	-7	-3	-5
Defender Reward	9	8	6	8	8	7	5	1
Defender Penalty	-5	-6	-4	-7	-5	-6	-3	-1

Table B.58: Reward structure 58

Doors	0	1	2	3	4	5	6	7
Subject Reward	7	3	8	4	2	10	7	4
Subject Penalty	-3	-4	-8	-3	-4	-10	-9	-2
Defender Reward	6	4	6	9	1	5	9	7
Defender Penalty	-3	-7	-8	-8	-1	-5	-7	-7

Table B.59: Reward structure 59

Doors	0	1	2	3	4	5	6	7
Subject Reward	6	10	2	6	7	4	9	10
Subject Penalty	-3	-7	-2	-4	-7	-3	-2	-10
Defender Reward	4	4	7	10	9	6	2	6
Defender Penalty	-10	-5	-8	-10	-6	-8	-4	-8

Table B.60: Reward structure 60

Doors	0	1	2	3	4	5	6	7
Subject Reward	8	9	10	6	1	6	10	8
Subject Penalty	-10	-3	-3	-9	-2	-4	-3	-3
Defender Reward	10	6	6	4	2	3	8	4
Defender Penalty	-9	-6	-9	-2	-5	-5	-10	-6

Table B.61: Reward structure 61

Doors	0	1	2	3	4	5	6	7
Subject Reward	4	7	6	4	1	9	8	6
Subject Penalty	-4	-6	-6	-8	-1	-8	-6	-6
Defender Reward	2	9	1	6	6	4	8	6
Defender Penalty	-3	-7	-1	-7	-7	-5	-10	-5

Table B.62: Reward structure 62

Doors	0	1	2	3	4	5	6	7
Subject Reward	7	6	1	5	7	10	6	6
Subject Penalty	-2	-7	-2	-8	-3	-4	-6	-2
Defender Reward	6	5	5	3	7	1	7	4
Defender Penalty	-9	-6	-4	-4	-8	-1	-8	-4

Table B.63: Reward structure 63

Doors	0	1	2	3	4	5	6	7
Subject Reward	2	8	6	6	4	1	3	8
Subject Penalty	-1	-10	-6	-8	-2	-2	-2	-8
Defender Reward	6	7	1	10	6	4	5	4
Defender Penalty	-7	-9	-3	-9	-6	-2	-6	-4

Table B.64: Reward structure 64

Doors	0	1	2	3	4	5	6	7
Subject Reward	3	4	5	1	2	3	2	2
Subject Penalty	-3	-4	-7	-3	-1	-7	-3	-4
Defender Reward	5	8	9	1	10	6	8	8
Defender Penalty	-4	-5	-8	-1	-7	-2	-5	-4

Table B.65: Reward structure 65

Doors	0	1	2	3	4	5	6	7
Subject Reward	4	1	4	1	5	5	6	4
Subject Penalty	-8	-1	-10	-7	-9	-6	-8	-7
Defender Reward	7	10	5	2	8	1	1	3
Defender Penalty	-7	-7	-4	-4	-4	-5	-2	-7

Table B.66: Reward structure 66

Doors	0	1	2	3	4	5	6	7
Subject Reward	10	1	7	8	7	5	1	7
Subject Penalty	-8	-7	-10	-6	-9	-7	-5	-6
Defender Reward	8	1	6	6	4	4	1	5
Defender Penalty	-8	-1	-8	-4	-6	-3	-3	-1

Table B.67: Reward structure 67

Doors	0	1	2	3	4	5	6	7
Subject Reward	9	9	9	4	10	5	8	6
Subject Penalty	-9	-6	-10	-6	-8	-3	-6	-4
Defender Reward	10	8	1	2	4	5	6	10
Defender Penalty	-7	-8	-2	-3	-6	-5	-6	-8

Table B.68: Reward structure 68

Doors	0	1	2	3	4	5	6	7
Subject Reward	10	7	8	4	6	4	1	10
Subject Penalty	-9	-1	-7	-7	-3	-2	-1	-4
Defender Reward	6	5	4	2	5	1	8	6
Defender Penalty	-5	-5	-8	-3	-8	-7	-9	-1

Table B.69: Reward structure 69

Doors	0	1	2	3	4	5	6	7
Subject Reward	5	8	9	6	1	6	10	2
Subject Penalty	-5	-5	-6	-10	-5	-9	-6	-5
Defender Reward	3	3	10	10	8	9	9	1
Defender Penalty	-3	-4	-7	-9	-5	-5	-6	-2

Table B.70: Reward structure 70

Doors	0	1	2	3	4	5	6	7
Subject Reward	7	1	5	6	6	4	1	4
Subject Penalty	-9	-1	-2	-4	-7	-3	-3	-2
Defender Reward	6	5	7	3	4	2	3	1
Defender Penalty	-7	-6	-10	-5	-1	-6	-4	-2

Table B.71: Reward structure 71

Doors	0	1	2	3	4	5	6	7
Subject Reward	8	10	5	5	1	5	6	10
Subject Penalty	-3	-8	-6	-7	-3	-7	-6	-6
Defender Reward	1	6	10	2	6	6	1	8
Defender Penalty	-4	-6	-8	-5	-5	-4	-5	-8

Table B.72: Reward structure 72

Doors	0	1	2	3	4	5	6	7
Subject Reward	4	8	3	8	7	7	10	8
Subject Penalty	-4	-6	-3	-6	-7	-4	-10	-9
Defender Reward	3	7	4	8	9	3	1	10
Defender Penalty	-5	-6	-5	-7	-7	-6	-6	-7

Table B.73: Reward structure 73

Doors	0	1	2	3	4	5	6	7
Subject Reward	8	1	2	5	5	2	3	8
Subject Penalty	-2	-2	-1	-4	-6	-3	-7	-4
Defender Reward	7	8	2	10	9	6	5	1
Defender Penalty	-6	-2	-5	-4	-7	-3	-6	-1

Table B.74: Reward structure 74

Doors	0	1	2	3	4	5	6	7
Subject Reward	2	8	8	6	3	6	9	1
Subject Penalty	-5	-7	-9	-7	-5	-2	-7	-1
Defender Reward	10	8	8	6	9	4	6	4
Defender Penalty	-10	-3	-8	-7	-7	-7	-1	-7

Table B.75: Reward structure 75

Doors	0	1	2	3	4	5	6	7
Subject Reward	7	9	3	7	10	5	5	1
Subject Penalty	-8	-5	-4	-9	-7	-5	-7	-1
Defender Reward	8	5	5	10	9	8	9	3
Defender Penalty	-5	-7	-6	-10	-6	-9	-6	-7

Table B.76: Reward structure 76

Doors	0	1	2	3	4	5	6	7
Subject Reward	6	6	7	8	1	9	9	7
Subject Penalty	-6	-9	-2	-8	-1	-8	-10	-5
Defender Reward	6	3	10	7	9	1	6	3
Defender Penalty	-8	-8	-5	-6	-7	-4	-3	-7

Table B.77: Reward structure 77

Doors	0	1	2	3	4	5	6	7
Subject Reward	8	6	1	10	9	5	4	8
Subject Penalty	-3	-5	-2	-9	-5	-2	-5	-7
Defender Reward	1	2	6	9	6	8	1	6
Defender Penalty	-3	-1	-6	-8	-2	-5	-5	-5

Table B.78: Reward structure 78

Doors	0	1	2	3	4	5	6	7
Subject Reward	6	7	1	5	5	6	4	5
Subject Penalty	-4	-3	-5	-8	-2	-1	-7	-5
Defender Reward	4	8	10	5	5	8	7	7
Defender Penalty	-3	-6	-5	-3	-3	-6	-10	-4

Table B.79: Reward structure 79

Doors	0	1	2	3	4	5	6	7
Subject Reward	1	5	5	10	4	5	6	3
Subject Penalty	-6	-5	-4	-6	-6	-1	-6	-5
Defender Reward	1	6	7	5	10	5	4	6
Defender Penalty	-6	-5	-8	-3	-4	-4	-2	-5

Table B.80: Reward structure 80

Doors	0	1	2	3	4	5	6	7
Subject Reward	2	1	7	7	1	6	3	9
Subject Penalty	-5	-1	-2	-6	-9	-9	-9	-10
Defender Reward	2	10	9	4	6	9	4	10
Defender Penalty	-1	-8	-9	-5	-10	-10	-3	-2

Table B.81: Reward structure 81

Doors	0	1	2	3	4	5	6	7
Subject Reward	5	6	5	7	8	6	5	5
Subject Penalty	-7	-5	-1	-3	-10	-7	-4	-8
Defender Reward	3	1	6	10	4	2	3	4
Defender Penalty	-5	-8	-5	-8	-3	-6	-4	-10

Table B.82: Reward structure 82

Doors	0	1	2	3	4	5	6	7
Subject Reward	7	6	1	8	6	4	9	9
Subject Penalty	-4	-2	-1	-2	-2	-2	-5	-7
Defender Reward	1	2	2	3	8	4	5	3
Defender Penalty	-5	-2	-7	-1	-8	-9	-10	-8

Table B.83: Reward structure 83

Doors	0	1	2	3	4	5	6	7
Subject Reward	7	8	6	4	6	9	1	2
Subject Penalty	-7	-5	-6	-10	-3	-6	-5	-1
Defender Reward	6	6	6	5	5	3	5	1
Defender Penalty	-7	-4	-5	-7	-1	-9	-6	-5

Table B.84: Reward structure 84

Doors	0	1	2	3	4	5	6	7
Subject Reward	5	1	6	7	2	6	4	6
Subject Penalty	-8	-1	-5	-2	-1	-2	-2	-5
Defender Reward	4	5	4	8	1	10	4	2
Defender Penalty	-8	-7	-5	-10	-7	-8	-7	-6

Table B.85: Reward structure 85

Doors	0	1	2	3	4	5	6	7
Subject Reward	9	10	9	1	2	6	9	7
Subject Penalty	-5	-1	-2	-5	-6	-3	-7	-9
Defender Reward	5	9	4	2	7	8	5	4
Defender Penalty	-3	-6	-1	-10	-3	-4	-2	-2

Table B.86: Reward structure 86

Doors	0	1	2	3	4	5	6	7
Subject Reward	1	6	4	4	7	6	7	6
Subject Penalty	-2	-5	-7	-1	-5	-2	-1	-3
Defender Reward	1	3	5	2	5	4	6	5
Defender Penalty	-1	-5	-8	-2	-7	-8	-4	-5

Table B.87: Reward structure 87

Doors	0	1	2	3	4	5	6	7
Subject Reward	3	10	3	5	2	4	9	1
Subject Penalty	-8	-7	-5	-3	-6	-10	-7	-2
Defender Reward	6	6	6	6	4	8	10	1
Defender Penalty	-5	-7	-5	-10	-4	-7	-7	-8

Table B.88: Reward structure 88

Doors	0	1	2	3	4	5	6	7
Subject Reward	1	7	8	8	9	8	4	7
Subject Penalty	-3	-6	-2	-7	-6	-1	-5	-4
Defender Reward	5	3	6	3	6	6	1	8
Defender Penalty	-7	-9	-5	-10	-4	-7	-8	-1

Table B.89: Reward structure 89

Doors	0	1	2	3	4	5	6	7
Subject Reward	9	3	1	5	4	6	1	7
Subject Penalty	-9	-7	-1	-5	-5	-9	-3	-4
Defender Reward	9	5	3	10	8	4	3	3
Defender Penalty	-8	-10	-9	-9	-9	-1	-5	-9

Table B.90: Reward structure 90

Doors	0	1	2	3	4	5	6	7
Subject Reward	8	9	6	6	5	5	1	7
Subject Penalty	-8	-3	-5	-5	-1	-3	-2	-4
Defender Reward	6	1	4	6	7	5	6	1
Defender Penalty	-3	-1	-6	-1	-8	-4	-2	-2

Table B.91: Reward structure 91

Doors	0	1	2	3	4	5	6	7
Subject Reward	8	1	9	7	8	9	10	9
Subject Penalty	-5	-2	-5	-2	-6	-5	-8	-9
Defender Reward	8	8	7	6	6	10	8	8
Defender Penalty	-7	-7	-2	-9	-10	-7	-3	-5

Table B.92: Reward structure 92

Doors	0	1	2	3	4	5	6	7
Subject Reward	4	7	5	9	6	1	5	4
Subject Penalty	-4	-3	-4	-1	-3	-5	-2	-7
Defender Reward	7	10	2	10	6	3	6	7
Defender Penalty	-4	-6	-4	-6	-10	-5	-5	-6

Table B.93: Reward structure 93

Doors	0	1	2	3	4	5	6	7
Subject Reward	10	3	6	5	1	8	1	6
Subject Penalty	-8	-5	-4	-8	-8	-3	-1	-2
Defender Reward	3	5	3	1	6	4	8	4
Defender Penalty	-9	-7	-1	-4	-2	-9	-6	-2

Table B.94: Reward structure 94

Doors	0	1	2	3	4	5	6	7
Subject Reward	6	1	8	9	10	8	6	3
Subject Penalty	-4	-7	-2	-8	-1	-6	-10	-4
Defender Reward	5	5	4	1	10	1	2	6
Defender Penalty	-8	-8	-5	-7	-8	-5	-7	-6

Table B.95: Reward structure 95

Doors	0	1	2	3	4	5	6	7
Subject Reward	6	7	8	7	6	8	5	6
Subject Penalty	-1	-10	-5	-9	-5	-6	-9	-5
Defender Reward	10	5	5	2	6	3	6	4
Defender Penalty	-7	-5	-2	-10	-9	-5	-10	-5

Table B.96: Reward structure 96

Doors	0	1	2	3	4	5	6	7
Subject Reward	8	4	2	3	5	10	5	4
Subject Penalty	-6	-8	-1	-2	-6	-6	-3	-10
Defender Reward	5	3	4	2	5	4	1	2
Defender Penalty	-7	-4	-7	-5	-1	-4	-6	-6

Table B.97: Reward structure 97

Doors	0	1	2	3	4	5	6	7
Subject Reward	2	4	4	1	9	6	8	5
Subject Penalty	-6	-2	-3	-1	-4	-6	-10	-6
Defender Reward	10	8	6	2	6	10	7	9
Defender Penalty	-4	-6	-9	-7	-1	-10	-4	-5

Table B.98: Reward structure 98

Doors	0	1	2	3	4	5	6	7
Subject Reward	7	4	4	4	8	8	1	10
Subject Penalty	-8	-4	-8	-1	-1	-6	-5	-6
Defender Reward	10	5	4	9	5	3	4	2
Defender Penalty	-10	-1	-5	-7	-5	-1	-4	-2

Table B.99: Reward structure 99

Doors	0	1	2	3	4	5	6	7
Subject Reward	6	6	1	7	2	7	4	6
Subject Penalty	-8	-3	-7	-9	-6	-5	-1	-6
Defender Reward	1	4	8	5	10	7	2	8
Defender Penalty	-6	-7	-8	-2	-7	-2	-9	-3

Table B.100: Reward structure 100

Doors	0	1	2	3	4	5	6	7
Subject Reward	6	1	4	10	1	1	2	6
Subject Penalty	-7	-10	-7	-1	-6	-4	-4	-7
Defender Reward	6	8	1	8	10	1	4	3
Defender Penalty	-6	-7	-4	-1	-7	-5	-7	-5

Table B.101: Reward structure 101

Doors	0	1	2	3	4	5	6	7
Subject Reward	2	1	9	10	7	1	6	6
Subject Penalty	-6	-8	-2	-4	-7	-7	-4	-3
Defender Reward	10	5	6	5	7	4	8	1
Defender Penalty	-3	-5	-8	-5	-5	-1	-4	-3

Table B.102: Reward structure 102

Doors	0	1	2	3	4	5	6	7
Subject Reward	7	10	2	9	9	1	1	8
Subject Penalty	-2	-3	-4	-4	-2	-5	-6	-3
Defender Reward	7	10	4	5	8	1	6	7
Defender Penalty	-2	-7	-6	-5	-10	-2	-7	-5

Table B.103: Reward structure 103

Doors	0	1	2	3	4	5	6	7
Subject Reward	4	6	5	8	10	4	10	8
Subject Penalty	-3	-5	-10	-4	-5	-3	-7	-4
Defender Reward	3	4	4	4	5	10	6	6
Defender Penalty	-5	-6	-6	-6	-6	-10	-10	-8

Table B.104: Reward structure 104

Doors	0	1	2	3	4	5	6	7
Subject Reward	4	8	4	10	8	1	6	5
Subject Penalty	-3	-6	-3	-6	-4	-7	-1	-5
Defender Reward	3	8	5	10	6	1	5	5
Defender Penalty	-1	-3	-3	-6	-4	-7	-7	-2

Table B.105: Reward structure 105

Doors	0	1	2	3	4	5	6	7
Subject Reward	2	4	7	9	9	8	1	10
Subject Penalty	-5	-2	-4	-6	-2	-4	-3	-3
Defender Reward	4	1	3	8	4	10	7	10
Defender Penalty	-9	-5	-3	-3	-9	-8	-5	-9

Table B.106: Reward structure 106

Doors	0	1	2	3	4	5	6	7
Subject Reward	3	10	5	8	7	8	6	1
Subject Penalty	-5	-3	-6	-4	-1	-2	-4	-7
Defender Reward	3	9	6	7	10	6	7	1
Defender Penalty	-7	-4	-4	-7	-8	-5	-1	-10

Table B.107: Reward structure 107

Doors	0	1	2	3	4	5	6	7
Subject Reward	7	9	3	2	4	7	5	5
Subject Penalty	-10	-1	-7	-3	-4	-8	-2	-4
Defender Reward	10	10	5	9	9	8	1	7
Defender Penalty	-7	-3	-3	-8	-7	-3	-8	-5

Table B.108: Reward structure 108

Appendix C: Strategies

	DOBSS	COBRA (0,2.5)	COBRA (1,2.5)	COBRA (.37,2.5)	COBRA (.03,2.5)	MAXIMIN
Door 1	0	.06	0	0	.06	.56
Door 2	.58	.63	.77	.64	.63	.53
Door 3	.45	.21	0	.23	.21	0
Door 4	.51	.58	.81	.63	.58	.48
Door 5	.56	.51	.70	.52	.51	.37
Door 6	0	.06	0	0	.06	0
Door 7	.61	.55	.69	.55	.55	.44
Door 8	.27	.36	0	.40	.36	.59

Table C.1: Mixed strategies for reward structure 1

	DOBSS	COBRA (0,2.5)	COBRA (1,2.5)	COBRA (.54,2.5)	COBRA (.41,2.5)	MAXIMIN
Door 1	.55	.57	1	.67	.61	.53
Door 2	.44	.55	0	.60	.56	.64
Door 3	.18	0	0	.05	.13	0
Door 4	.67	.53	1	.62	.57	.48
Door 5	0	0	0	0	0	0
Door 6	.18	.31	0	.05	.13	.27
Door 7	.64	.61	1	.69	.65	.58
Door 8	.30	.41	0	.29	.32	.48

Table C.2: Mixed strategies for reward structure 2

	DOBSS	COBRA (0,2.5)	COBRA (1,2.5)	COBRA (.75,2.5)	COBRA (.25,2.5)	MAXIMIN
Door 1	.58	.52	1	.71	.55	.48
Door 2	.43	.41	0	.60	.46	.35
Door 3	.09	0	0	0	0	0
Door 4	.65	.55	1	.70	.57	.52
Door 5	0	.17	0	0	0	.47
Door 6	.24	.26	0	0	.33	.17
Door 7	.62	.52	1	.68	.54	.49
Door 8	.35	.53	0	.28	.51	.48

Table C.3: Mixed strategies for reward structure 3

	DOBSS	COBRA (0,2.5)	COBRA (1,2.5)	COBRA (.75,2.5)	COBRA (.25,2.5)	MAXIMIN
Door 1	.58	.58	1	.68	.58	.58
Door 2	.43	.43	0	.56	.43	.43
Door 3	.09	.09	0	0	.09	.09
Door 4	.65	.65	1	.73	.65	.65
Door 5	0	0	0	0	0	0
Door 6	.24	.24	0	0	.24	.24
Door 7	.62	.62	1	.70	.62	.62
Door 8	.35	.35	0	.32	.35	.35

Table C.4: Mixed strategies for reward structure 4

	DOBSS	BRQR	MATCH
Door 1	0.49118	0.56923	0.57388
Door 2	0.52917	0.57955	0.55339
Door 3	0.15	0.18303	0.18394
Door 4	0.35667	0.20853	0.2389
Door 5	0.435	0.5053	0.48196
Door 6	0.59445	0.47195	0.43158
Door 7	0.37353	0.29801	0.29979
Door 8	0.070004	0.18439	0.23657

Table C.5: Mixed strategies for reward structure 5

	DOBSS	BRQR	MATCH
Door 1	0.42135	0.53544	0.55473
Door 2	0.7784	0.51819	0.46059
Door 3	0.083799	0.20815	0.2212
Door 4	0.47092	0.35989	0.32412
Door 5	0.64326	0.63863	0.56254
Door 6	0.00050859	0.15929	0.23964
Door 7	0.60037	0.58041	0.51197
Door 8	0.0013986	0	0.12521

Table C.6: Mixed strategies for reward structure 6

	DOBSS	BRQR	MATCH
Door 1	0.53446	0.36491	0.39159
Door 2	0.36549	0.42707	0.40101
Door 3	0.1206	0.19989	0.20146
Door 4	0.24824	0.35737	0.38017
Door 5	0.0603	0.12692	0.15881
Door 6	0.72027	0.72366	0.66188
Door 7	0.3103	0.42603	0.45685
Door 8	0.64034	0.37415	0.34822

Table C.7: Mixed strategies for reward structure 7

	DOBSS	BRQR	MATCH
Door 1	0.21734	0.34625	0.44487
Door 2	0.36809	0.33067	0.27811
Door 3	0.19017	0.29879	0.32545
Door 4	0.44244	0.43861	0.39942
Door 5	0.047399	0.19562	0.28346
Door 6	0.57857	0.61677	0.58471
Door 7	0.69017	0.35675	0.32151
Door 8	0.46581	0.41654	0.36249

Table C.8: Mixed strategies for reward structure 8

	BRQR	MATCH
Door 1	0.031979	0.30562
Door 2	0.4777	0.43852
Door 3	0.43569	0.40342
Door 4	0.40438	0.37115
Door 5	0.45842	0.40929
Door 6	0.49791	0.43379
Door 7	0.32578	0.29326
Door 8	0.36814	0.34495

Table C.9: Mixed strategies for reward structure 9

	BRQR	MATCH
Door 1	0.44102	0.3917
Door 2	0.48892	0.44037
Door 3	0.0083501	0.30092
Door 4	0.39495	0.33395
Door 5	0.49699	0.42901
Door 6	0.21231	0.23583
Door 7	0.46688	0.42343
Door 8	0.49058	0.44478

Table C.10: Mixed strategies for reward structure 10

	BRQR	MATCH
Door 1	0.46554	0.41412
Door 2	0.33212	0.3801
Door 3	0.44026	0.3434
Door 4	0.29086	0.19882
Door 5	0.31655	0.46278
Door 6	0.50599	0.41732
Door 7	0.038925	0.15579
Door 8	0.60976	0.62767

Table C.11: Mixed strategies for reward structure 11

	BRQR	MATCH
Door 1	0.4173	0.37247
Door 2	0.50216	0.48697
Door 3	0.20484	0.24417
Door 4	0.16726	0.36237
Door 5	0.46177	0.41689
Door 6	0.59813	0.62879
Door 7	0.27382	0.24417
Door 8	0.37471	0.24417

Table C.12: Mixed strategies for reward structure 12

	BRQR	MATCH
Door 1	0.37698	0.32346
Door 2	0.4177	0.36151
Door 3	0.14489	0.35729
Door 4	0.59455	0.57837
Door 5	0.32782	0.30269
Door 6	0.20829	0.26215
Door 7	0.4157	0.3918
Door 8	0.51406	0.42274

Table C.13: Mixed strategies for reward structure 13

	BRQR	MATCH
Door 1	0.54347	0.50963
Door 2	0.32934	0.31286
Door 3	0.54583	0.52715
Door 4	0.42059	0.42435
Door 5	0.11906	0.18401
Door 6	0.42951	0.39819
Door 7	0.31543	0.28801
Door 8	0.29676	0.3558

Table C.14: Mixed strategies for reward structure 14

	BRQR	MATCH
Door 1	0.46362	0.46477
Door 2	0.49376	0.47437
Door 3	0.37271	0.33801
Door 4	0.40476	0.33801
Door 5	0.36136	0.33801
Door 6	0.062786	0.24363
Door 7	0.39218	0.3786
Door 8	0.44883	0.4246

Table C.15: Mixed strategies for reward structure 15

	BRQR	MATCH
Door 1	0.32818	0.30559
Door 2	0.4332	0.40911
Door 3	0.2943	0.28128
Door 4	0.33182	0.34213
Door 5	0.46742	0.46776
Door 6	0.45422	0.41073
Door 7	0.27545	0.3667
Door 8	0.4154	0.4167

Table C.16: Mixed strategies for reward structure 16

	BRQR	MATCH
Door 1	0.44183	0.4142
Door 2	0.35812	0.29807
Door 3	0.44112	0.39231
Door 4	0.18538	0.39819
Door 5	0.49288	0.45266
Door 6	0.35812	0.36686
Door 7	0.43102	0.37965
Door 8	0.29154	0.29807

Table C.17: Mixed strategies for reward structure 17

	BRQR	MATCH
Door 1	0.4458	0.47245
Door 2	0.43025	0.41434
Door 3	0.22804	0.3134
Door 4	0.41189	0.40532
Door 5	0.37431	0.34804
Door 6	0.36769	0.33505
Door 7	0.30076	0.30842
Door 8	0.44124	0.40297

Table C.18: Mixed strategies for reward structure 18

	BRQR	MATCH
Door 1	0.37952	0.38886
Door 2	0.13566	0.15621
Door 3	0.4243	0.4259
Door 4	0.42282	0.38886
Door 5	0.48221	0.47724
Door 6	0.45763	0.40621
Door 7	0.31536	0.34371
Door 8	0.3825	0.41301

Table C.19: Mixed strategies for reward structure 19

	BRQR	MATCH
Door 1	0.60075	0.51171
Door 2	0.45821	0.40342
Door 3	0.096945	0.411
Door 4	0.14126	0.19279
Door 5	0.44011	0.39626
Door 6	0.41305	0.35922
Door 7	0.45771	0.40342
Door 8	0.39197	0.32219

Table C.20: Mixed strategies for reward structure 20

	BRQR	MATCH
Door 1	0.43642	0.39444
Door 2	0.54295	0.48029
Door 3	0.54423	0.47343
Door 4	0.10386	0.34444
Door 5	0.38067	0.32592
Door 6	0.13658	0.21605
Door 7	0.47641	0.43827
Door 8	0.37888	0.32716

Table C.21: Mixed strategies for reward structure 21

	BRQR	MATCH
Door 1	0.16548	0.21549
Door 2	0.28342	0.35472
Door 3	0.3309	0.32378
Door 4	0.53527	0.50337
Door 5	0.44191	0.4092
Door 6	0.36485	0.33963
Door 7	0.4468	0.4271
Door 8	0.43137	0.42671

Table C.22: Mixed strategies for reward structure 22

	BRQR	MATCH
Door 1	0.27491	0.24258
Door 2	0.19842	0.22825
Door 3	0.40263	0.3591
Door 4	0.33373	0.30569
Door 5	0.67307	0.62982
Door 6	0.18927	0.33351
Door 7	0.39357	0.40759
Door 8	0.5344	0.49347

Table C.23: Mixed strategies for reward structure 23

	BRQR	MATCH
Door 1	0.38948	0.3638
Door 2	0.36987	0.34672
Door 3	0.34544	0.38547
Door 4	0.064649	0.13966
Door 5	0.49677	0.45742
Door 6	0.4578	0.46229
Door 7	0.39784	0.36393
Door 8	0.47815	0.48071

Table C.24: Mixed strategies for reward structure 24

	BRQR	MATCH
Door 1	0.43008	0.40137
Door 2	0.62192	0.60998
Door 3	0.31931	0.31832
Door 4	0.34255	0.33198
Door 5	0.25743	0.25635
Door 6	0.46621	0.47998
Door 7	0.359	0.34427
Door 8	0.2035	0.25776

Table C.25: Mixed strategies for reward structure 25

	BRQR	MATCH
Door 1	0.35358	0.3495
Door 2	0.33268	0.33476
Door 3	0.289	0.33548
Door 4	0.38236	0.38279
Door 5	0.4286	0.39511
Door 6	0.35396	0.37045
Door 7	0.44104	0.40884
Door 8	0.41879	0.42308

Table C.26: Mixed strategies for reward structure 26

	BRQR	MATCH
Door 1	0.50705	0.49056
Door 2	0.11007	0.22798
Door 3	0.38917	0.38679
Door 4	0.41173	0.39799
Door 5	0.51401	0.39708
Door 6	0.35927	0.36399
Door 7	0.28498	0.30617
Door 8	0.42372	0.42943

Table C.27: Mixed strategies for reward structure 27

	BRQR	MATCH
Door 1	0.49902	0.44633
Door 2	0.13916	0.20752
Door 3	0.47985	0.3833
Door 4	0.45827	0.42354
Door 5	0.26543	0.26414
Door 6	0.39678	0.36012
Door 7	0.41923	0.38678
Door 8	0.34225	0.52827

Table C.28: Mixed strategies for reward structure 28

	BRQR	MATCH
Door 1	0.3575	0.34615
Door 2	0.47803	0.44322
Door 3	0.27061	0.29486
Door 4	0.40718	0.38461
Door 5	0.41499	0.40468
Door 6	0.43451	0.42081
Door 7	0.44495	0.4523
Door 8	0.19222	0.25338

Table C.29: Mixed strategies for reward structure 29

	BRQR	MATCH
Door 1	0.43063	0.41385
Door 2	0	0.11656
Door 3	0.45197	0.41718
Door 4	0.47723	0.43919
Door 5	0.45622	0.40978
Door 6	0.31171	0.3575
Door 7	0.41129	0.42375
Door 8	0.46095	0.4222

Table C.30: Mixed strategies for reward structure 30

	BRQR	MATCH
Door 1	0.34639	0.30989
Door 2	0.4706	0.43594
Door 3	0.46481	0.38594
Door 4	0.53714	0.48918
Door 5	0.44229	0.39631
Door 6	0.12073	0.30989
Door 7	0.30095	0.37187
Door 8	0.3171	0.30099

Table C.31: Mixed strategies for reward structure 31

	BRQR	MATCH
Door 1	0.31577	0.30768
Door 2	0.47374	0.49997
Door 3	0.32059	0.30768
Door 4	0.069525	0.07689
Door 5	0.43191	0.43332
Door 6	0.39255	0.41174
Door 7	0.52478	0.47998
Door 8	0.47112	0.48274

Table C.32: Mixed strategies for reward structure 32

	BRQR	MATCH
Door 1	0.50682	0.47617
Door 2	0.37494	0.33568
Door 3	0.37317	0.37297
Door 4	0.068584	0.22612
Door 5	0.39025	0.35976
Door 6	0.37813	0.37885
Door 7	0.44578	0.41206
Door 8	0.46232	0.4384

Table C.33: Mixed strategies for reward structure 33

	BRQR	MATCH
Door 1	0.61702	0.55101
Door 2	0.41251	0.43955
Door 3	0.18253	0.14284
Door 4	0.37428	0.33926
Door 5	0.46092	0.40134
Door 6	0.050871	0.24282
Door 7	0.58041	0.53201
Door 8	0.32146	0.35118

Table C.34: Mixed strategies for reward structure 34

	BRQR	MATCH
Door 1	0.28671	0.31944
Door 2	0.47916	0.47037
Door 3	0.31598	0.33862
Door 4	0.30974	0.30917
Door 5	0.34622	0.32444
Door 6	0.50902	0.5037
Door 7	0.44492	0.43386
Door 8	0.30825	0.30041

Table C.35: Mixed strategies for reward structure 35

	BRQR	MATCH
Door 1	0.27227	0.30417
Door 2	0.45538	0.4164
Door 3	0.30964	0.28786
Door 4	0.46744	0.43081
Door 5	0.40012	0.38733
Door 6	0.40898	0.43081
Door 7	0.67703	0.64168
Door 8	0.0091507	0.10095

Table C.36: Mixed strategies for reward structure 36

	BRQR	MATCH
Door 1	0.28789	0.26184
Door 2	0.51608	0.4771
Door 3	0.45695	0.39638
Door 4	0.15973	0.41724
Door 5	0.51787	0.47511
Door 6	0.55388	0.49722
Door 7	0.5076	0.47511
Door 8	0	0

Table C.37: Mixed strategies for reward structure 37

	BRQR	MATCH
Door 1	0.56642	0.54596
Door 2	0.21141	0.21373
Door 3	0.58391	0.57991
Door 4	0.56425	0.52785
Door 5	0.22608	0.23984
Door 6	0.38485	0.391
Door 7	0.1877	0.20924
Door 8	0.27538	0.29247

Table C.38: Mixed strategies for reward structure 38

	BRQR	MATCH
Door 1	0.46496	0.43254
Door 2	0.29043	0.25794
Door 3	0.49292	0.47061
Door 4	0.34785	0.33618
Door 5	0.40875	0.40989
Door 6	0.26522	0.42786
Door 7	0.43867	0.4056
Door 8	0.2912	0.25938

Table C.39: Mixed strategies for reward structure 39

	BRQR	MATCH
Door 1	0.28949	0.30123
Door 2	0.5489	0.48743
Door 3	0.42605	0.33667
Door 4	0.55799	0.49012
Door 5	0.1915	0.24884
Door 6	0.37062	0.42907
Door 7	0.23361	0.35771
Door 8	0.38184	0.34893

Table C.40: Mixed strategies for reward structure 40

	BRQR	MATCH
Door 1	0.45055	0.40924
Door 2	0.4554	0.4028
Door 3	0.51544	0.46143
Door 4	0.35551	0.33993
Door 5	0.04993	0.22352
Door 6	0.44562	0.40924
Door 7	0.39275	0.36392
Door 8	0.3348	0.38991

Table C.41: Mixed strategies for reward structure 41

	BRQR	MATCH
Door 1	0.35292	0.34806
Door 2	0.31637	0.34585
Door 3	0.46813	0.41988
Door 4	0.40765	0.37991
Door 5	0.1051	0.27979
Door 6	0.48265	0.41321
Door 7	0.42004	0.3877
Door 8	0.44713	0.42561

Table C.42: Mixed strategies for reward structure 42

	BRQR	MATCH
Door 1	0.37769	0.34431
Door 2	0.41657	0.38299
Door 3	0.46925	0.43362
Door 4	0.21833	0.38977
Door 5	0.37363	0.34299
Door 6	0.39878	0.38977
Door 7	0.42408	0.38977
Door 8	0.32168	0.32678

Table C.43: Mixed strategies for reward structure 43

	BRQR	MATCH
Door 1	0.41457	0.38161
Door 2	0.43524	0.41184
Door 3	0.38047	0.40598
Door 4	0.42115	0.39756
Door 5	0.19463	0.27464
Door 6	0.26696	0.28442
Door 7	0.42835	0.40398
Door 8	0.45862	0.43998

Table C.44: Mixed strategies for reward structure 44

	BRQR	MATCH
Door 1	0.42465	0.40677
Door 2	0.418	0.39719
Door 3	0.29717	0.3181
Door 4	0.52491	0.53737
Door 5	0.07314	0.11353
Door 6	0.38538	0.35371
Door 7	0.47161	0.44516
Door 8	0.40513	0.42817

Table C.45: Mixed strategies for reward structure 45

	BRQR	MATCH
Door 1	0.49474	0.42774
Door 2	0.18393	0.34892
Door 3	0.43068	0.38921
Door 4	0.10799	0.32254
Door 5	0.4949	0.42774
Door 6	0.39571	0.32254
Door 7	0.42113	0.3599
Door 8	0.47092	0.40141

Table C.46: Mixed strategies for reward structure 46

	BRQR	MATCH
Door 1	0.48397	0.42384
Door 2	0.31742	0.29966
Door 3	0.44684	0.39986
Door 4	0.18649	0.32828
Door 5	0.46564	0.41225
Door 6	0.47309	0.41722
Door 7	0.47019	0.39971
Door 8	0.15635	0.31919

Table C.47: Mixed strategies for reward structure 47

	BRQR	MATCH
Door 1	0.51531	0.47046
Door 2	0.39659	0.34343
Door 3	0.16129	0.36075
Door 4	0.44093	0.38417
Door 5	0.43653	0.39688
Door 6	0.43674	0.36635
Door 7	0.43931	0.38417
Door 8	0.1733	0.29377

Table C.48: Mixed strategies for reward structure 48

	BRQR	MATCH
Door 1	0.40117	0.39587
Door 2	0.29662	0.29433
Door 3	0.37767	0.39408
Door 4	0.42459	0.40977
Door 5	0.45114	0.43756
Door 6	0.36206	0.38399
Door 7	0.22716	0.25399
Door 8	0.45958	0.4304

Table C.49: Mixed strategies for reward structure 49

	BRQR	MATCH
Door 1	0.3721	0.33997
Door 2	0.12554	0.31133
Door 3	0.27868	0.24456
Door 4	0.47045	0.42967
Door 5	0.55222	0.52342
Door 6	0.3385	0.37994
Door 7	0.35658	0.31214
Door 8	0.50593	0.45896

Table C.50: Mixed strategies for reward structure 50

	BRQR	MATCH
Door 1	0.39568	0.35102
Door 2	0.35783	0.36239
Door 3	0.5387	0.50334
Door 4	0.40083	0.37229
Door 5	0.40797	0.36698
Door 6	0.32015	0.38744
Door 7	0.25174	0.33946
Door 8	0.3271	0.31709

Table C.51: Mixed strategies for reward structure 51

	BRQR	MATCH
Door 1	0.43382	0.40477
Door 2	0.16767	0.23283
Door 3	0.40267	0.38097
Door 4	0.51126	0.46769
Door 5	0.37703	0.40954
Door 6	0.5263	0.47303
Door 7	0.13348	0.29253
Door 8	0.44777	0.33864

Table C.52: Mixed strategies for reward structure 52

	BRQR	MATCH
Door 1	0.37649	0.36842
Door 2	0.11258	0.16666
Door 3	0.40139	0.38095
Door 4	0.26258	0.40909
Door 5	0.4276	0.3913
Door 6	0.43927	0.4
Door 7	0.53929	0.47619
Door 8	0.44081	0.4074

Table C.53: Mixed strategies for reward structure 53

	BRQR	MATCH
Door 1	0.43673	0.38351
Door 2	0.51469	0.4709
Door 3	0.44994	0.40165
Door 4	0.36181	0.33287
Door 5	0.4091	0.36067
Door 6	0.096859	0.37942
Door 7	0.26558	0.26114
Door 8	0.46528	0.40985

Table C.54: Mixed strategies for reward structure 54

	BRQR	MATCH
Door 1	0.42172	0.3958
Door 2	0.56386	0.5633
Door 3	0	0
Door 4	0.4175	0.4033
Door 5	0.35006	0.3633
Door 6	0.29089	0.27216
Door 7	0.52383	0.53426
Door 8	0.43214	0.46788

Table C.55: Mixed strategies for reward structure 55

	BRQR	MATCH
Door 1	0.48509	0.51224
Door 2	0.50724	0.46557
Door 3	0.18479	0.26486
Door 4	0.20747	0.27081
Door 5	0.47907	0.38832
Door 6	0.40002	0.35675
Door 7	0.32164	0.37861
Door 8	0.41469	0.36284

Table C.56: Mixed strategies for reward structure 56

	BRQR	MATCH
Door 1	0.40729	0.36784
Door 2	0.32416	0.35175
Door 3	0.50221	0.46784
Door 4	0.47732	0.46901
Door 5	0.49512	0.44574
Door 6	0.019537	0.18969
Door 7	0.32669	0.29313
Door 8	0.44767	0.415

Table C.57: Mixed strategies for reward structure 57

	BRQR	MATCH
Door 1	0.28892	0.27302
Door 2	0.48979	0.47843
Door 3	0.38726	0.37703
Door 4	0.53269	0.47227
Door 5	0.39921	0.38444
Door 6	0.36113	0.3567
Door 7	0.24213	0.27983
Door 8	0.29886	0.37828

Table C.58: Mixed strategies for reward structure 58

	BRQR	MATCH
Door 1	0.38324	0.4046
Door 2	0.37355	0.42708
Door 3	0.47918	0.45625
Door 4	0.44904	0.40364
Door 5	0	0.085923
Door 6	0.44161	0.42291
Door 7	0.39608	0.36523
Door 8	0.4773	0.43437

Table C.59: Mixed strategies for reward structure 59

	BRQR	MATCH
Door 1	0.50092	0.48407
Door 2	0.38745	0.38975
Door 3	0.17799	0.27019
Door 4	0.42356	0.37112
Door 5	0.31399	0.28047
Door 6	0.34873	0.33969
Door 7	0.43053	0.47844
Door 8	0.41684	0.38628

Table C.60: Mixed strategies for reward structure 60

	BRQR	MATCH
Door 1	0.3745	0.34318
Door 2	0.4709	0.44574
Door 3	0.56405	0.52492
Door 4	0.17164	0.17609
Door 5	0	0.16978
Door 6	0.3798	0.3721
Door 7	0.55047	0.50638
Door 8	0.48864	0.4618

Table C.61: Mixed strategies for reward structure 61

	BRQR	MATCH
Door 1	0.33695	0.33335
Door 2	0.43828	0.39081
Door 3	0.15886	0.30954
Door 4	0.3579	0.33334
Door 5	0.30991	0.35557
Door 6	0.464	0.4359
Door 7	0.53148	0.47917
Door 8	0.40261	0.36233

Table C.62: Mixed strategies for reward structure 62

	BRQR	MATCH
Door 1	0.5911	0.52094
Door 2	0.39276	0.35428
Door 3	0	0.12522
Door 4	0.30511	0.27513
Door 5	0.52632	0.46011
Door 6	0.30131	0.46892
Door 7	0.43305	0.38899
Door 8	0.45034	0.40642

Table C.63: Mixed strategies for reward structure 63

	BRQR	MATCH
Door 1	0.45781	0.42106
Door 2	0.45493	0.43344
Door 3	0.37887	0.42106
Door 4	0.41701	0.38597
Door 5	0.46953	0.42983
Door 6	0	0.081882
Door 7	0.43749	0.42106
Door 8	0.38435	0.40571

Table C.64: Mixed strategies for reward structure 64

	BRQR	MATCH
Door 1	0.45396	0.45077
Door 2	0.4555	0.41722
Door 3	0.46995	0.44006
Door 4	0.0056021	0.2936
Door 5	0.59619	0.43808
Door 6	0.26599	0.26453
Door 7	0.41032	0.37565
Door 8	0.34249	0.32009

Table C.65: Mixed strategies for reward structure 65

	BRQR	MATCH
Door 1	0.39175	0.37822
Door 2	0.54356	0.35967
Door 3	0.29889	0.29712
Door 4	0.19516	0.27383
Door 5	0.30175	0.30129
Door 6	0.49048	0.51963
Door 7	0.32958	0.40198
Door 8	0.44882	0.46827

Table C.66: Mixed strategies for reward structure 66

	BRQR	MATCH
Door 1	0.54516	0.49945
Door 2	0.069616	0.098137
Door 3	0.48612	0.45101
Door 4	0.48393	0.45756
Door 5	0.49695	0.46082
Door 6	0.40539	0.36744
Door 7	0.20291	0.29814
Door 8	0.30994	0.36744

Table C.67: Mixed strategies for reward structure 67

	BRQR	MATCH
Door 1	0.37854	0.36109
Door 2	0.46088	0.43994
Door 3	0.29125	0.34719
Door 4	0.20145	0.24254
Door 5	0.45986	0.45136
Door 6	0.36941	0.36878
Door 7	0.42045	0.40916
Door 8	0.41816	0.37993

Table C.68: Mixed strategies for reward structure 68

	BRQR	MATCH
Door 1	0.4422	0.36283
Door 2	0.55464	0.43804
Door 3	0.49131	0.44018
Door 4	0.24028	0.1803
Door 5	0.52049	0.44931
Door 6	0.45387	0.49177
Door 7	0.00014699	0.30973
Door 8	0.29706	0.32785

Table C.69: Mixed strategies for reward structure 69

	BRQR	MATCH
Door 1	0.4041	0.39791
Door 2	0.53658	0.51833
Door 3	0.47882	0.44896
Door 4	0.40901	0.3819
Door 5	0.18271	0.22982
Door 6	0.36088	0.32299
Door 7	0.47326	0.46344
Door 8	0.15464	0.23666

Table C.70: Mixed strategies for reward structure 70

	BRQR	MATCH
Door 1	0.44097	0.38996
Door 2	0.29028	0.33144
Door 3	0.61224	0.51286
Door 4	0.51033	0.4616
Door 5	0.15146	0.23937
Door 6	0.51439	0.48725
Door 7	0.14127	0.20988
Door 8	0.33906	0.36764

Table C.71: Mixed strategies for reward structure 71

	BRQR	MATCH
Door 1	0.52781	0.54695
Door 2	0.43288	0.42504
Door 3	0.38384	0.33625
Door 4	0.3605	0.35533
Door 5	0.10065	0.18341
Door 6	0.28404	0.26142
Door 7	0.43291	0.43062
Door 8	0.47738	0.46098

Table C.72: Mixed strategies for reward structure 72

	BRQR	MATCH
Door 1	0.30431	0.33387
Door 2	0.39066	0.38304
Door 3	0.24532	0.28946
Door 4	0.40868	0.3911
Door 5	0.3691	0.34473
Door 6	0.46677	0.4671
Door 7	0.46137	0.45711
Door 8	0.35378	0.33359

Table C.73: Mixed strategies for reward structure 73

	BRQR	MATCH
Door 1	0.57141	0.53758
Door 2	0.10586	0.10494
Door 3	0.42112	0.53643
Door 4	0.38014	0.32019
Door 5	0.43545	0.38386
Door 6	0.25774	0.2403
Door 7	0.34384	0.35068
Door 8	0.48443	0.52602

Table C.74: Mixed strategies for reward structure 74

	BRQR	MATCH
Door 1	0.32703	0.34091
Door 2	0.36262	0.31556
Door 3	0.46221	0.40014
Door 4	0.44843	0.39249
Door 5	0.34614	0.30019
Door 6	0.61849	0.53709
Door 7	0.27591	0.31325
Door 8	0.15916	0.40036

Table C.75: Mixed strategies for reward structure 75

	BRQR	MATCH
Door 1	0.35316	0.31418
Door 2	0.53429	0.49219
Door 3	0.33999	0.32206
Door 4	0.4298	0.38325
Door 5	0.40742	0.39991
Door 6	0.45228	0.39989
Door 7	0.34306	0.28878
Door 8	0.14	0.39975

Table C.76: Mixed strategies for reward structure 76

	BRQR	MATCH
Door 1	0.43721	0.40539
Door 2	0.39897	0.40539
Door 3	0.41051	0.35583
Door 4	0.40814	0.36345
Door 5	0.063105	0.25222
Door 6	0.46385	0.43364
Door 7	0.31285	0.305
Door 8	0.50536	0.47909

Table C.77: Mixed strategies for reward structure 77

	BRQR	MATCH
Door 1	0.56592	0.53324
Door 2	0.28511	0.28561
Door 3	0.091549	0.26657
Door 4	0.47013	0.41663
Door 5	0.33376	0.36357
Door 6	0.44054	0.34993
Door 7	0.37489	0.3999
Door 8	0.43811	0.38456

Table C.78: Mixed strategies for reward structure 78

	BRQR	MATCH
Door 1	0.41023	0.4141
Door 2	0.49533	0.45999
Door 3	0.11596	0.19237
Door 4	0.2946	0.28761
Door 5	0.40593	0.40265
Door 6	0.53505	0.47808
Door 7	0.38748	0.42999
Door 8	0.35542	0.33522

Table C.79: Mixed strategies for reward structure 79

	BRQR	MATCH
Door 1	0.17808	0.35364
Door 2	0.41652	0.37862
Door 3	0.50074	0.45629
Door 4	0.43239	0.45629
Door 5	0.3001	0.24796
Door 6	0.51275	0.4634
Door 7	0.33493	0.33061
Door 8	0.3245	0.31321

Table C.80: Mixed strategies for reward structure 80

	BRQR	MATCH
Door 1	0.14671	0.18259
Door 2	0.41762	0.39129
Door 3	0.63166	0.54911
Door 4	0.52872	0.49208
Door 5	0.24677	0.37792
Door 6	0.45888	0.43606
Door 7	0.26453	0.25399
Door 8	0.30511	0.31696

Table C.81: Mixed strategies for reward structure 81

	BRQR	MATCH
Door 1	0.30979	0.30961
Door 2	0.47901	0.50961
Door 3	0.40944	0.36424
Door 4	0.44928	0.39972
Door 5	0.26792	0.28769
Door 6	0.38576	0.3901
Door 7	0.31966	0.32451
Door 8	0.37914	0.41452

Table C.82: Mixed strategies for reward structure 82

	BRQR	MATCH
Door 1	0.48061	0.41884
Door 2	0.31624	0.26002
Door 3	0	0.28366
Door 4	0.21482	0.2943
Door 5	0.48607	0.38001
Door 6	0.44407	0.42738
Door 7	0.55418	0.4869
Door 8	0.50401	0.4489

Table C.83: Mixed strategies for reward structure 83

	BRQR	MATCH
Door 1	0.47145	0.41113
Door 2	0.43511	0.39567
Door 3	0.41751	0.35219
Door 4	0.3277	0.31156
Door 5	0.24379	0.27336
Door 6	0.60648	0.55928
Door 7	0.17665	0.2412
Door 8	0.32131	0.45561

Table C.84: Mixed strategies for reward structure 84

	BRQR	MATCH
Door 1	0.36963	0.34642
Door 2	0.084023	0.26146
Door 3	0.37142	0.33302
Door 4	0.54667	0.4689
Door 5	0.28798	0.42368
Door 6	0.45824	0.37156
Door 7	0.44427	0.39179
Door 8	0.43776	0.40318

Table C.85: Mixed strategies for reward structure 85

	BRQR	MATCH
Door 1	0.48604	0.4257
Door 2	0.57309	0.51405
Door 3	0.47912	0.46033
Door 4	0.14453	0.46474
Door 5	0.12215	0.13141
Door 6	0.43166	0.35073
Door 7	0.41821	0.36371
Door 8	0.3452	0.28933

Table C.86: Mixed strategies for reward structure 86

	BRQR	MATCH
Door 1	0	0
Door 2	0.42932	0.42063
Door 3	0.3466	0.37467
Door 4	0.25777	0.33245
Door 5	0.48519	0.458
Door 6	0.57703	0.5496
Door 7	0.45554	0.444
Door 8	0.44855	0.42063

Table C.87: Mixed strategies for reward structure 87

	BRQR	MATCH
Door 1	0.27885	0.24926
Door 2	0.50591	0.48279
Door 3	0.32766	0.28862
Door 4	0.58417	0.52015
Door 5	0.224	0.21773
Door 6	0.32474	0.29254
Door 7	0.42292	0.4086
Door 8	0.33176	0.54031

Table C.88: Mixed strategies for reward structure 88

	BRQR	MATCH
Door 1	0	0.22296
Door 2	0.47572	0.4627
Door 3	0.49254	0.40797
Door 4	0.49537	0.48455
Door 5	0.40848	0.3427
Door 6	0.57638	0.48034
Door 7	0.34615	0.42041
Door 8	0.20536	0.17837

Table C.89: Mixed strategies for reward structure 89

	BRQR	MATCH
Door 1	0.44926	0.3942
Door 2	0.38558	0.39188
Door 3	0.37214	0.4855
Door 4	0.45608	0.37231
Door 5	0.44403	0.37681
Door 6	0.10908	0.18985
Door 7	0.16599	0.23308
Door 8	0.61784	0.55639

Table C.90: Mixed strategies for reward structure 90

	BRQR	MATCH
Door 1	0.35937	0.34247
Door 2	0.46894	0.54012
Door 3	0.47832	0.45532
Door 4	0.22474	0.25343
Door 5	0.58586	0.50294
Door 6	0.42154	0.38598
Door 7	0	0.051061
Door 8	0.46123	0.46869

Table C.91: Mixed strategies for reward structure 91

	BRQR	MATCH
Door 1	0.45596	0.40372
Door 2	0	0.23912
Door 3	0.31766	0.31757
Door 4	0.55963	0.51267
Door 5	0.4946	0.4768
Door 6	0.45358	0.39691
Door 7	0.33961	0.32083
Door 8	0.37895	0.33239

Table C.92: Mixed strategies for reward structure 92

	BRQR	MATCH
Door 1	0.30274	0.27302
Door 2	0.46241	0.39182
Door 3	0.39765	0.41249
Door 4	0.51947	0.46874
Door 5	0.55123	0.52749
Door 6	0.024132	0.22767
Door 7	0.44453	0.3993
Door 8	0.29784	0.29947

Table C.93: Mixed strategies for reward structure 93

	BRQR	MATCH
Door 1	0.58607	0.55635
Door 2	0.38565	0.38453
Door 3	0.25527	0.33504
Door 4	0.38271	0.3717
Door 5	0.060333	0.040625
Door 6	0.66425	0.61211
Door 7	0.28311	0.29316
Door 8	0.38262	0.40647

Table C.94: Mixed strategies for reward structure 94

	BRQR	MATCH
Door 1	0.44436	0.4128
Door 2	0.050399	0.21401
Door 3	0.5048	0.44707
Door 4	0.47632	0.45977
Door 5	0.51961	0.46532
Door 6	0.45607	0.42472
Door 7	0.33367	0.33977
Door 8	0.21478	0.23654

Table C.95: Mixed strategies for reward structure 95

	BRQR	MATCH
Door 1	0.45507	0.37696
Door 2	0.3195	0.29804
Door 3	0.2552	0.30235
Door 4	0.41958	0.46596
Door 5	0.43017	0.42488
Door 6	0.42757	0.41123
Door 7	0.3291	0.36823
Door 8	0.36381	0.35235

Table C.96: Mixed strategies for reward structure 96

	BRQR	MATCH
Door 1	0.50416	0.45682
Door 2	0.28621	0.2567
Door 3	0.40689	0.41981
Door 4	0.39385	0.40644
Door 5	0.11433	0.16925
Door 6	0.44542	0.45322
Door 7	0.53501	0.52515
Door 8	0.31413	0.31261

Table C.97: Mixed strategies for reward structure 97

	BRQR	MATCH
Door 1	0.21249	0.16662
Door 2	0.48702	0.38328
Door 3	0.52795	0.4848
Door 4	0.15425	0.51505
Door 5	0.3449	0.38328
Door 6	0.49103	0.42705
Door 7	0.39813	0.3333
Door 8	0.38422	0.30662

Table C.98: Mixed strategies for reward structure 98

	BRQR	MATCH
Door 1	0.45856	0.42273
Door 2	0.20426	0.19968
Door 3	0.31772	0.3236
Door 4	0.51708	0.41883
Door 5	0.59705	0.56818
Door 6	0.33064	0.37753
Door 7	0.084073	0.19968
Door 8	0.49063	0.48978

Table C.99: Mixed strategies for reward structure 99

	BRQR	MATCH
Door 1	0.44257	0.44742
Door 2	0.56432	0.51979
Door 3	0.18584	0.26649
Door 4	0.29319	0.27808
Door 5	0.25455	0.25583
Door 6	0.29668	0.30456
Door 7	0.64627	0.64974
Door 8	0.31657	0.27808

Table C.100: Mixed strategies for reward structure 100

	BRQR	MATCH
Door 1	0.49679	0.4084
Door 2	0.20353	0.23884
Door 3	0.4179	0.38812
Door 4	0.52119	0.4605
Door 5	0.24172	0.25875
Door 6	0.22153	0.38272
Door 7	0.38888	0.42411
Door 8	0.50847	0.43857

Table C.101: Mixed strategies for reward structure 101

	BRQR	MATCH
Door 1	0.19414	0.16762
Door 2	0.16265	0.2379
Door 3	0.65623	0.6208
Door 4	0.57062	0.56333
Door 5	0.43565	0.40462
Door 6	0.01059	0.040001
Door 7	0.41914	0.38727
Door 8	0.55098	0.57846

Table C.102: Mixed strategies for reward structure 102

	BRQR	MATCH
Door 1	0.40715	0.35545
Door 2	0.53811	0.47994
Door 3	0.2162	0.33738
Door 4	0.55561	0.49557
Door 5	0.64485	0.56545
Door 6	0	0.044228
Door 7	0.11716	0.2699
Door 8	0.52093	0.45209

Table C.103: Mixed strategies for reward structure 103

	BRQR	MATCH
Door 1	0.25257	0.28932
Door 2	0.35074	0.34951
Door 3	0.24863	0.25359
Door 4	0.42363	0.42453
Door 5	0.41793	0.43614
Door 6	0.36486	0.34592
Door 7	0.48558	0.46484
Door 8	0.45606	0.43614

Table C.104: Mixed strategies for reward structure 104

	BRQR	MATCH
Door 1	0.24216	0.25119
Door 2	0.37358	0.35052
Door 3	0.34624	0.31754
Door 4	0.46241	0.4301
Door 5	0.48172	0.44378
Door 6	0.15178	0.3602
Door 7	0.6298	0.56648
Door 8	0.31232	0.28018

Table C.105: Mixed strategies for reward structure 105

	BRQR	MATCH
Door 1	0.21946	0.34945
Door 2	0.38255	0.41575
Door 3	0.40698	0.3523
Door 4	0.32616	0.30727
Door 5	0.65287	0.58288
Door 6	0.47732	0.39963
Door 7	0	0.12431
Door 8	0.53467	0.46841

Table C.106: Mixed strategies for reward structure 106

	BRQR	MATCH
Door 1	0.26473	0.36808
Door 2	0.45255	0.40867
Door 3	0.31198	0.26788
Door 4	0.51445	0.44713
Door 5	0.55379	0.44713
Door 6	0.53651	0.45835
Door 7	0.26508	0.20141
Door 8	0.10091	0.40134

Table C.107: Mixed strategies for reward structure 107

	BRQR	MATCH
Door 1	0.38035	0.34287
Door 2	0.45712	0.41989
Door 3	0.19809	0.20319
Door 4	0.25018	0.34807
Door 5	0.3774	0.36073
Door 6	0.35367	0.29452
Door 7	0.57872	0.66609
Door 8	0.40447	0.36464

Table C.108: Mixed strategies for reward structure 108

Appendix D: Expected Rewards for COBRA Experiments

	DOBSS	COBRA (0,2.5)	COBRA (1,2.5)	COBRA (.37,2.5)	COBRA (.03,2.5)	MAXIMIN	UNIFORM
Door 1	-5	-4.58	-5	-5	-4.60	-1.62	-2.78
Door 2	-.96	-.42	1.35	-.29	-.36	-1.62	-3.56
Door 3	.35	-.35	-1	-.29	-.36	-1	.11
Door 4	-1.37	-.79	1.35	-.29	-.73	-1.62	-2.67
Door 5	.05	-.35	1.35	-.29	-.36	-1.62	-1.67
Door 6	-1	-.86	-1	-1	-.86	-1	-.26
Door 7	.38	-.35	1.35	-.29	-.36	-1.62	-2.56
Door 8	-4.56	-3.68	-7	-3.32	-3.67	-1.62	-3.67

Table D.1: Expected rewards for reward structure 1

	DOBSS	COBRA (0,2.5)	COBRA (1,2.5)	COBRA (.54,2.5)	COBRA (.41,2.5)	MAXIMIN	UNIFORM
Door 1	-1.32	-1.09	4	.09	-.57	-1.63	-3.56
Door 2	-4.21	-2.81	-10	-2.12	-2.69	-1.63	-5.19
Door 3	-.62	-1	-1	-.89	-.72	-1	-.26
Door 4	.79	-1.09	5	.09	-.57	-1.63	-3.19
Door 5	-1	-1	-1	-1	-1	-1	-.26
Door 6	-2.06	-1.44	-3	-2.72	-2.31	-1.63	-1.15
Door 7	-.64	-1.09	5	.09	-.57	-1.63	-5.08
Door 8	-2.88	-2.12	-5	-2.93	-2.75	-1.63	-2.41

Table D.2: Expected rewards for reward structure 2

	DOBSS	COBRA	COBRA	COBRA	COBRA	MAXIMIN	UNIFORM
		(0,2.5)	(1,2.5)	(.75,2.5)	(.25,2.5)		
Door 1	-0.92	-1.66	4	.62	-1.31	-2.12	-3.56
Door 2	-1.51	-1.66	-5	-0.16	-1.31	-2.12	-2.04
Door 3	-0.80	-1	-1	-1	-1	-1	-0.26
Door 4	-.21	-1.66	5	.62	-1.31	-2.12	-4.45
Door 5	-5	-3.92	-5	-5	-5	-2.12	-2.78
Door 6	-1.76	-1.66	-3	-3	-1.31	-2.12	-1.15
Door 7	-0.26	-1.66	5	.62	-1.31	-2.12	-3.82
Door 8	-3.16	-1.74	-6	-3.75	-1.87	-2.12	-3.04

Table D.3: Expected rewards for reward structure 3

	DOBSS	COBRA	COBRA	COBRA	COBRA	MAXIMIN	UNIFORM
		(0,2.5)	(1,2.5)	(.75,2.5)	(.25,2.5)		
Door 1	-1.51	-1.51	3	-.50	-1.51	-1.51	-3.93
Door 2	-1.51	-1.51	-5	-.50	-1.51	-1.51	-2.04
Door 3	-1.51	-1.51	-2	-2	-1.51	-1.51	-0.15
Door 4	-1.51	-1.51	3	-.50	-1.51	-1.51	-5.19
Door 5	-1	-1	-1	-1	-1	-1	.48
Door 6	-1.51	-1.51	-3	-3	-1.51	-1.51	-0.78
Door 7	-1.51	-1.51	3	-.50	-1.51	-1.51	-4.56
Door 8	-1.51	-1.51	-4	-1.75	-1.51	-1.51	-1.41

Table D.4: Expected rewards for reward structure 4

Appendix E: Expected Response Percentages for COBRA Experiment

Structure One	Unobserved	5	20	Unlimited
DOBSS	20%	7.5%	17.5%	12.5%
COBRA(0, ϵ)	65%	65%	65%	70%
COBRA(α , ϵ)	57.5%	92.5%	72.5%	70%
COBRA(C, ϵ)	92.5%	92.5%	87.5%	95%
MAXIMIN	100%	100%	100%	100%
Structure Two				
DOBSS	27.5%	25%	12.5%	10%
COBRA(0, ϵ)	62.5%	65%	40%	55%
COBRA(α , ϵ)	62.5%	57.5%	47.5%	55%
COBRA(C, ϵ)	62.5%	57.5%	55%	47.5%
MAXIMIN	100%	100%	100%	100%
Structure Three				
DOBSS	20%	20%	20%	25%
COBRA(0, ϵ)	75%	72.5%	62.5%	60%
COBRA(α , ϵ)	50%	47.5%	67.5%	60%
COBRA(C, ϵ)	50%	47.5%	20%	25%
MAXIMIN	100%	100%	100%	100%
Structure Four				
DOBSS	100%	92.5%	87.5%	85%
COBRA(0, ϵ)	100%	92.5%	87.5%	85%
COBRA(α , ϵ)	42.5%	52.5%	82.5%	85%
COBRA(C, ϵ)	52.5%	52.5%	25%	35%
MAXIMIN	100%	100%	100%	100%

Table E.1: Percentage of times follower chose an *expected strategy*

Appendix F: Strategies for varying α in COBRA($\alpha,2.5$)

	Door 1	Door 2	Door 3	Door 4	Door 5	Door 6	Door 7	Door 8
$\alpha = .00$.069	.631	.213	.578	.515	.069	.553	.368
$\alpha = .05$.062	.635	.208	.592	.513	.062	.552	.372
$\alpha = .10$.056	.634	.203	.608	.512	.056	.550	.377
$\alpha = .15$.051	.632	.198	.621	.510	.051	.549	.384
$\alpha = .20$.050	.632	.194	.620	.509	.050	.548	.394
$\alpha = .25$.049	.630	.190	.619	.507	.049	.547	.406
$\alpha = .30$.046	.630	.187	.618	.506	.046	.546	.418
$\alpha = .35$.005	.640	.227	.631	.520	.005	.556	.414
$\alpha = .40$.000	.643	.242	.636	.525	.000	.560	.391
$\alpha = .45$.000	.647	.256	.641	.529	.000	.564	.361
$\alpha = .50$.000	.651	.272	.646	.535	.000	.568	.325
$\alpha = .55$.000	.656	.292	.653	.542	.000	.573	.282
$\alpha = .60$.000	.662	.318	.661	.550	.000	.579	.227
$\alpha = .65$.000	.671	.350	.672	.561	.000	.587	.156
$\alpha = .70$.000	.681	.393	.686	.575	.000	.598	.063
$\alpha = .75$.000	.689	.423	.696	.585	.000	.605	.000
$\alpha = .80$.000	.689	.423	.696	.585	.000	.605	.000
$\alpha = .85$.000	.689	.423	.696	.585	.000	.605	.000
$\alpha = .90$.000	.689	.423	.696	.585	.000	.605	.000
$\alpha = .95$.000	.731	.227	.752	.641	.000	.647	.000
$\alpha = 1.00$.000	.779	.000	.817	.706	.000	.696	.000

Table F.1: α -variations for reward structure 1

	Door 1	Door 2	Door 3	Door 4	Door 5	Door 6	Door 7	Door 8
$\alpha = .00$.575	.553	.000	.530	.000	.311	.618	.410
$\alpha = .05$.579	.555	.000	.535	.000	.300	.622	.405
$\alpha = .10$.584	.557	.006	.539	.000	.287	.625	.399
$\alpha = .15$.587	.557	.026	.542	.000	.268	.628	.389
$\alpha = .20$.591	.556	.048	.546	.000	.248	.631	.377
$\alpha = .25$.595	.555	.074	.549	.000	.224	.634	.365
$\alpha = .30$.600	.554	.103	.554	.000	.198	.637	.350
$\alpha = .35$.606	.554	.136	.559	.000	.167	.642	.334
$\alpha = .40$.617	.560	.139	.569	.000	.139	.650	.322
$\alpha = .45$.634	.573	.114	.585	.000	.114	.663	.314
$\alpha = .50$.654	.589	.084	.604	.000	.084	.678	.304
$\alpha = .55$.679	.609	.046	.627	.000	.046	.697	.292
$\alpha = .60$.711	.634	.000	.656	.000	.000	.720	.277
$\alpha = .65$.730	.633	.000	.674	.000	.000	.735	.225
$\alpha = .70$.757	.632	.000	.698	.000	.000	.755	.156
$\alpha = .75$.793	.631	.000	.732	.000	.000	.782	.059
$\alpha = .80$.828	.597	.000	.765	.000	.000	.809	.000
$\alpha = .85$.863	.503	.000	.797	.000	.000	.635	.000
$\alpha = .90$.933	.316	.000	.861	.000	.000	.887	.000
$\alpha = .95$	1.00	.000	.000	1.00	.062	.000	.937	.000
$\alpha = 1.00$	1.00	.000	.000	1.00	.000	.000	1.00	.000

Table F.2: α -variations for reward structure 2

	Door 1	Door 2	Door 3	Door 4	Door 5	Door 6	Door 7	Door 8
$\alpha = .00$.527	.416	.000	.555	.179	.266	.523	.531
$\alpha = .05$.532	.423	.000	.559	.151	.277	.527	.529
$\alpha = .10$.537	.431	.000	.563	.119	.290	.532	.526
$\alpha = .15$.543	.440	.000	.568	.083	.304	.537	.523
$\alpha = .20$.550	.450	.000	.573	.043	.320	.542	.520
$\alpha = .25$.557	.460	.000	.579	.000	.337	.549	.516
$\alpha = .30$.559	.464	.000	.580	.000	.342	.550	.502
$\alpha = .35$.562	.468	.000	.583	.000	.345	.553	.487
$\alpha = .40$.570	.481	.000	.590	.000	.320	.560	.476
$\alpha = .45$.581	.496	.000	.598	.000	.290	.569	.464
$\alpha = .50$.593	.514	.000	.607	.000	.254	.579	.450
$\alpha = .55$.607	.536	.000	.619	.000	.210	.592	.432
$\alpha = .60$.626	.564	.000	.634	.000	.155	.608	.410
$\alpha = .65$.650	.600	.000	.653	.000	.085	.628	.381
$\alpha = .70$.687	.615	.000	.683	.000	.001	.660	.352
$\alpha = .75$.719	.604	.000	.708	.000	.000	.687	.280
$\alpha = .80$.766	.587	.000	.746	.000	.000	.728	.171
$\alpha = .85$.842	.556	.000	.807	.000	.000	.293	.000
$\alpha = .90$.908	.381	.000	.860	.000	.000	.850	.000
$\alpha = .95$	1.00	.071	.000	1.00	.000	.000	.928	.000
$\alpha = 1.00$	1.00	.000	.000	1.00	.000	.000	1.00	.000

Table F.3: α -variations for reward structure 3

	Door 1	Door 2	Door 3	Door 4	Door 5	Door 6	Door 7	Door 8
$\alpha = .00$.589	.435	.096	.652	.000	.247	.623	.354
$\alpha = .05$.589	.435	.096	.652	.000	.247	.623	.354
$\alpha = .10$.589	.435	.096	.652	.000	.247	.623	.354
$\alpha = .15$.589	.435	.096	.652	.000	.247	.623	.354
$\alpha = .20$.589	.435	.096	.652	.000	.247	.623	.354
$\alpha = .25$.589	.435	.096	.652	.000	.247	.623	.354
$\alpha = .30$.589	.435	.096	.652	.000	.247	.623	.354
$\alpha = .35$.589	.435	.096	.652	.000	.247	.623	.354
$\alpha = .40$.589	.435	.096	.652	.000	.247	.623	.354
$\alpha = .45$.589	.435	.096	.652	.000	.247	.623	.354
$\alpha = .50$.589	.435	.096	.652	.000	.247	.623	.354
$\alpha = .55$.600	.451	.010	.662	.000	.268	.634	.372
$\alpha = .60$.609	.462	.000	.669	.000	.231	.641	.285
$\alpha = .65$.624	.482	.000	.681	.000	.146	.655	.409
$\alpha = .70$.651	.520	.000	.705	.000	.048	.680	.393
$\alpha = .75$.681	.561	.000	.730	.000	.000	.707	.320
$\alpha = .80$.712	.604	.000	.756	.000	.000	.736	.190
$\alpha = .85$.787	.477	.000	.819	.000	.000	.804	.010
$\alpha = .90$.852	.406	.000	.875	.000	.000	.865	.000
$\alpha = .95$	1.00	.000	.000	1.00	.000	.000	1.00	.000
$\alpha = 1.00$	1.00	.000	.000	1.00	.000	.000	1.00	.000

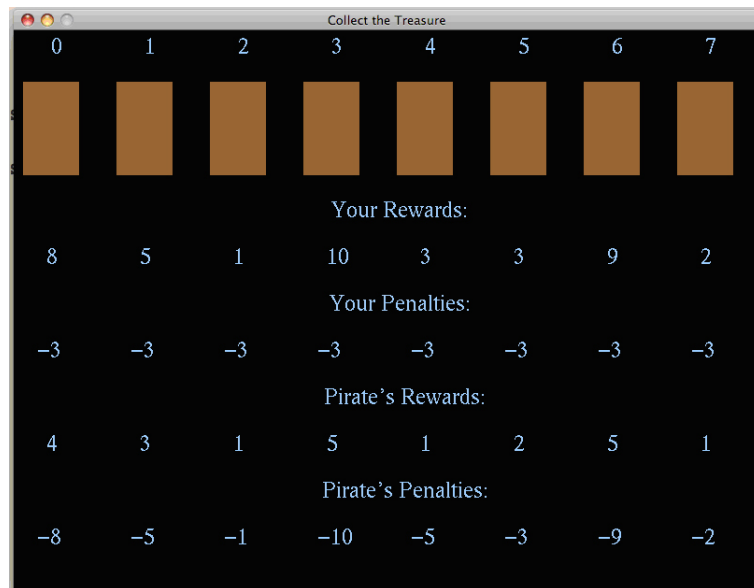
Table F.4: α -variations for reward structure 4

Appendix G: Experimental Instructions

G.1 Material for COBRA Experiments

You will play 14 unique and distinct instances of the following game. Please note that each instance is neither related to nor involves any of the other instances of the game. No memory will be maintained of either the pirates' actions or your choices. We will describe below how one instance of the game unfolds. Remember you will play 14 instances of the game.

Pirate Game: *These may not be the rewards you will see



0	1	2	3	4	5	6	7
8	5	1	10	3	3	9	2
-3	-3	-3	-3	-3	-3	-3	-3
4	3	1	5	1	2	5	1
-8	-5	-1	-10	-5	-3	-9	-2

Figure G.1: Game Interface

Description: As you were boating around one day you stumbled across an island inhabited by pirates and quickly discovered that they were storing mass amounts of treasure on the island. Knowing that this treasure was stolen from hard working people you thought it would only be fair to try and get some of this hard earned money back from the pirates. After some observation over several days you have found the following:

Layout:

- There are eight doors in total which lead to the pirates' treasure.

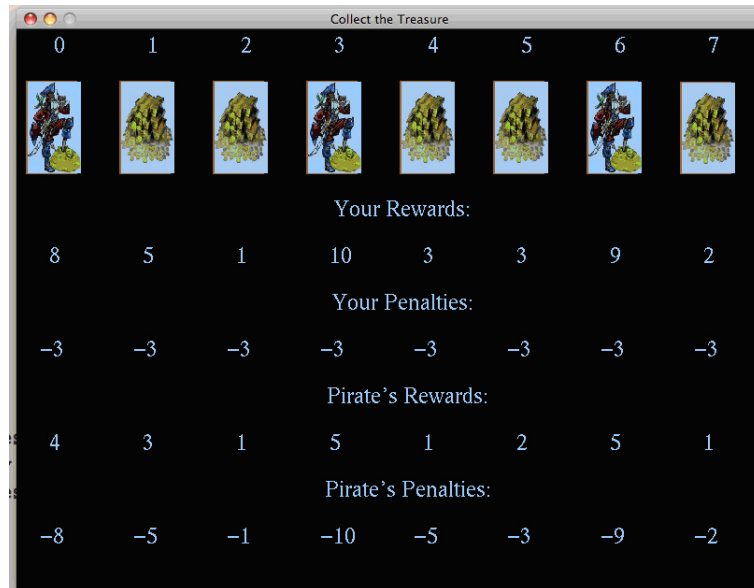


Figure G.2: Single Observation

- At any given time three of the eight doors will be guarded by pirates as shown in Figure Two.
- Each of the eight doors has a different amount of treasure behind it and thus is worth different values to both you and the pirates. The rewards for each of you are given as shown in both figures*. Both you and the pirate are aware of each others rewards and penalties. If you succeed then you get your reward and the pirate incurs their penalty. If you are caught then you incur your penalty and the pirate receives their reward.
- So initially, you see a picture as shown in Figure 1, where you don't know the pirates' position. When you make your choice, the pirates' positions will be revealed. In the above example, if you chose door 0 to steal from, you will get caught, incurring a penalty of -3, while the pirate wins a reward of 4. If you had instead chosen to steal from door 1, you would have obtained a reward of 5.
- You can only attempt to steal from a single door. Once you have made your choice you will be informed whether you got caught or not.

Rules:

- You will be given a specific number of observations to attempt and learn the pirates' strategy. Each observation consists of the doors opening to show you the locations of the pirates (i.e. which doors the pirates are guarding), and the pirates may change which doors they guard from day to day.
- Once your observations are complete you will be given an unlimited amount of time to choose a door to try and steal from. In essence, having observed the pirates for some number of days, you now have to make your move and steal from a particular door.

- Once a door is chosen you will see whether you got away with the treasure or got caught and the appropriate points will either be added or deducted from your overall score.

Conclusion:

- At the end of the twelve individual games your total score will decide how much money you earn. Each point you earn is worth 15 cents. The pirates' points have no bearing on how much money you will get.
- You will begin the game with 8 free dollars. If you have a negative score you will lose 15 cents per negative point up to \$3. You will be guaranteed to leave with at least \$5 no matter how negative your score is.

If you have any questions feel free to ask

G.2 Material for MATCH Experiments

G.2.1 Obvious Games

Gates	1	2	3	4	5	6	7	8
Subject Reward	10	1	9	2	3	10	2	4
Subject Penalty	-1	-5	-9	-2	-5	-8	-5	-3
Defender Reward	5	6	2	8	4	2	1	4
Defender Penalty	-2	-3	-2	-2	-3	-3	-3	-2
Defender Strategy	.05	.30	.50	.35	.30	.65	.35	.50

Table G.1: Dummy structure 1

Gates	1	2	3	4	5	6	7	8
Subject Reward	2	9	4	10	3	2	5	10
Subject Penalty	-5	-8	-9	-10	-5	-8	-5	-1
Defender Reward	5	6	2	8	4	2	1	4
Defender Penalty	-2	-3	-2	-2	-3	-3	-3	-2
Defender Strategy	.30	.70	.25	.60	.25	.35	.50	.05


Table G.2: Dummy structure 2

G.3 Experiment Instructions

Instructions


How is the game played?

In this game there are 8 gates which you will be able to choose from



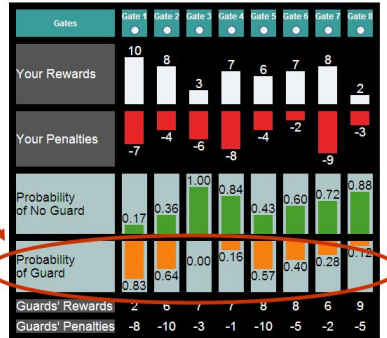
Gates	Gate 1	Gate 2	Gate 3	Gate 4	Gate 5	Gate 6	Gate 7	Gate 8
Your Rewards	10	8	3	7	6	7	8	2
Your Penalties	-7	-4	-6	-8	-4	-2	-9	-3
Probability of No Guard	0.17	0.36	1.00	0.84	0.43	0.60	0.72	0.88
Probability of Guard	0.83	0.64	0.00	0.16	0.57	0.40	0.28	0.12
Guards' Rewards	2	6	7	7	8	8	6	9
Guards' Penalties	-8	-10	-3	-1	-10	-5	-2	-5

These 8 gates are protected by 3 guards

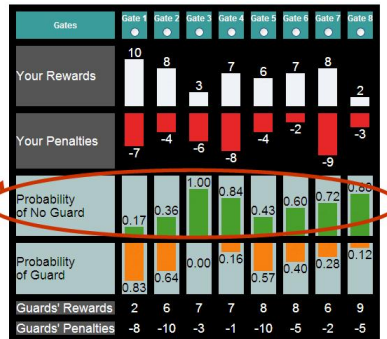


Gates	Gate 1	Gate 2	Gate 3	Gate 4	Gate 5	Gate 6	Gate 7	Gate 8
Your Rewards	10	8	3	7	6	7	8	2
Your Penalties	-7	-4	-6	-8	-4	-2	-9	-3
Probability of No Guard	0.17	0.36	1.00	0.84	0.43	0.60	0.72	0.88
Probability of Guard	0.83	0.64	0.00	0.16	0.57	0.40	0.28	0.12
Guards' Rewards	2	6	7	7	8	8	6	9
Guards' Penalties	-8	-10	-3	-1	-10	-5	-2	-5

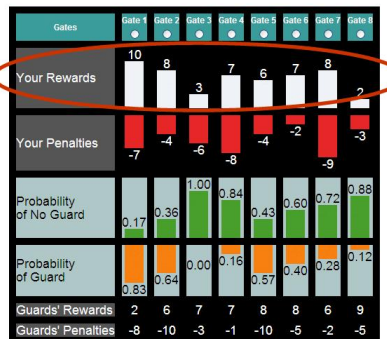
The 3 guards have agreed on a patrolling strategy where they will be present at each door with a certain probability



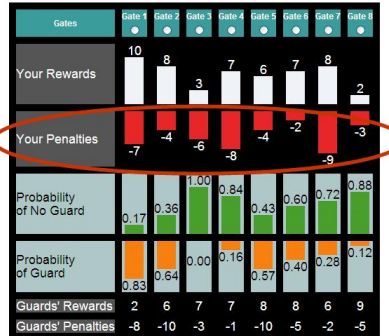
Conversely, this means there will be no guard at this gate with the opposite probability (e.g., a guard on gate 1 with .83 and no guard with .17)



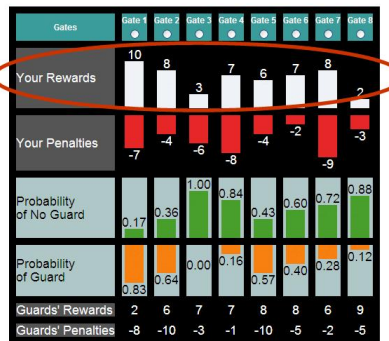
If you choose a gate and there is no guard you will receive a reward



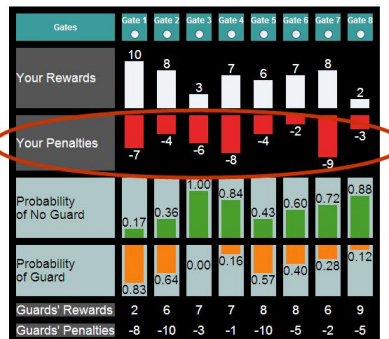
Conversely, if you choose a gate and there is a guard you will receive a penalty



For each reward point you earn we will pay you a bonus \$0.15



Conversely, for each penalty point we will deduct \$0.15 (Note: if your final score is negative we will not deduct from the base HIT payment)



Notice that the guards also have rewards and penalties

Gates	Gate 1	Gate 2	Gate 3	Gate 4	Gate 5	Gate 6	Gate 7	Gate 8
Your Rewards	10	8	3	7	6	7	8	2
Your Penalties	-7	-4	-6	-8	-4	-2	-9	-3
Probability of No Guard	0.17	0.36	1.00	0.84	0.43	0.60	0.72	0.88
Probability of Guard	0.83	0.64	0.00	0.16	0.57	0.40	0.28	0.12
Guards' Rewards	2	6	7	7	8	8	6	9
Guards' Penalties	-8	-10	-3	-1	-10	-5	-2	-5

This does not affect the outcome of your results, but if you get a reward they get a penalty, else vice versa

Gates	Gate 1	Gate 2	Gate 3	Gate 4	Gate 5	Gate 6	Gate 7	Gate 8
Your Rewards	10	8	3	7	6	7	8	2
Your Penalties	-7	-4	-6	-8	-4	-2	-9	-3
Probability of No Guard	0.17	0.36	1.00	0.84	0.43	0.60	0.72	0.88
Probability of Guard	0.83	0.64	0.00	0.16	0.57	0.40	0.28	0.12
Guards' Rewards	2	6	7	7	8	8	6	9
Guards' Penalties	-8	-10	-3	-1	-10	-5	-2	-5

This information is only provided so you are aware of why they may have taken this particular patrolling strategy

Gates	Gate 1	Gate 2	Gate 3	Gate 4	Gate 5	Gate 6	Gate 7	Gate 8
Your Rewards	10	8	3	7	6	7	8	2
Your Penalties	-7	-4	-6	-8	-4	-2	-9	-3
Probability of No Guard	0.17	0.36	1.00	0.84	0.43	0.60	0.72	0.88
Probability of Guard	0.83	0.64	0.00	0.16	0.57	0.40	0.28	0.12
Guards' Rewards	2	6	7	7	8	8	6	9
Guards' Penalties	-8	-10	-3	-1	-10	-5	-2	-5

You are going to play a number of game instances and in each game instance you can only select a single gate to score points

Gates	Gate 1	Gate 2	Gate 3	Gate 4	Gate 5	Gate 6	Gate 7	Gate 8
Your Rewards	10	8	3	7	6	7	8	2
Your Penalties	-7	-4	-6	-8	-4	-2	-9	-3
Probability of No Guard	0.17	0.36	1.00	0.84	0.43	0.60	0.72	0.88
Probability of Guard	0.83	0.64	0.00	0.16	0.57	0.40	0.28	0.12
Guards' Rewards	2	6	7	7	8	8	6	9
Guards' Penalties	-8	-10	-3	-1	-10	-5	-2	-5

Each game is unique, it will have a new set of targets guarded by a new set of guards who have chosen a different patrolling strategy

Gates	Gate 1	Gate 2	Gate 3	Gate 4	Gate 5	Gate 6	Gate 7	Gate 8
Your Rewards	3	7	3	9	2	7	8	
Your Penalties	-4	-8	-5	-8	-9	-4	-1	-6
Probability of No Guard	0.78	0.58	0.79	1.00	0.37	0.50	0.48	
Probability of Guard	0.22	0.42	0.21	0.00	0.63	0.50	0.52	
Guards' Rewards	5	9	10	2	10	4	8	8
Guards' Penalties	-10	-4	-9	-3	-10	-10	-2	-5

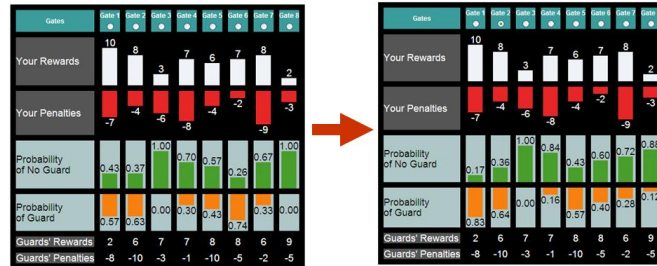
Gates	Gate 1	Gate 2	Gate 3	Gate 4	Gate 5	Gate 6	Gate 7	Gate 8
Your Rewards	10	8	3	7	6	7	8	2
Your Penalties	-7	-4	-6	-8	-4	-2	-9	-3
Probability of No Guard	0.17	0.36	1.00	0.84	0.43	0.60	0.72	0.88
Probability of Guard	0.83	0.64	0.00	0.16	0.57	0.40	0.28	0.12
Guards' Rewards	2	6	7	7	8	8	6	9
Guards' Penalties	-8	-10	-3	-1	-10	-5	-2	-5

Even if the target values appear the same notice that because it is a new set of targets and guards the patrolling strategy has still changed

Gates	Gate 1	Gate 2	Gate 3	Gate 4	Gate 5	Gate 6	Gate 7	Gate 8
Your Rewards	10	8	3	7	6	7	8	2
Your Penalties	-7	-4	-6	-8	-4	-2	-9	-3
Probability of No Guard	0.43	0.37	1.00	0.70	0.57	0.26	0.67	1.00
Probability of Guard	0.57	0.63	0.00	0.30	0.43	0.74	0.33	0.00
Guards' Rewards	2	6	7	7	8	8	6	9
Guards' Penalties	-8	-10	-3	-1	-10	-5	-2	-5

Gates	Gate 1	Gate 2	Gate 3	Gate 4	Gate 5	Gate 6	Gate 7	Gate 8
Your Rewards	10	8	3	7	6	7	8	2
Your Penalties	-7	-4	-6	-8	-4	-2	-9	-3
Probability of No Guard	0.17	0.36	1.00	0.84	0.43	0.60	0.72	0.88
Probability of Guard	0.83	0.64	0.00	0.16	0.57	0.40	0.28	0.12
Guards' Rewards	2	6	7	7	8	8	6	9
Guards' Penalties	-8	-10	-3	-1	-10	-5	-2	-5

It is up to you to decide which gate is best against each set of guards



You will begin observing each patrolling strategy before you have to make a decision and you will be given immediate feedback at the end of each game on whether you succeeded or not. However, at the very end of all games we will only select 5 games at random from all the games you played to determine your bonus payment. Your bonus payment will be the score you received for those 5 randomly selected games so do your best on each game!