Game-Theoretic Models for Interdependent Security: Modeling, Computing, and Learning

A Dissertation presented

by

Hau Chan

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

Stony Brook University

August 2015

Stony Brook University

The Graduate School

Hau Chan

We, the dissertation committee for the above candidate for the

Doctor of Philosophy degree, hereby recommend

acceptance of this dissertation

Luis E. Ortiz - Dissertation Advisor
Assistant Professor of Computer Science, Stony Brook University

Jie Gao - Chairperson of Defense
Associate Professor of Computer Science, Stony Brook University

Jing Chen - 3rd Inside Member
Assistant Professor of Computer Science, Stony Brook University

Vincent Conitzer - External Member
Sally Dalton Robinson Professor of Computer Science
and Professor of Economics, Duke University

This dissertation is accepted by the Graduate School

Charles Taber
Dean of the Graduate School

Abstract of the Dissertation

Game-Theoretic Models for Interdependent Security: Modeling, Computing, and Learning

by

Hau Chan

Doctor of Philosophy

in

Computer Science

Stony Brook University

2015

Due to an increase number of attacks by hackers and terrorists, there has been quite a bit of recent research activity in the general area of game-theoretic models for terrorism settings that aim to understand the behavior of the attackers and the attackers' targets. My thesis is centered on introducing, studying, and applying several game-theoretic models to security.

In particular, my doctoral thesis consists of the following components: (1) designing increasingly more realistic variants of defense games; (2) studying computational questions in defense games such as equilibria computation and computational implications of equilibria characterizations, (3) designing efficient algorithms and effective heuristics for defense problems; and (4) designing and applying new machine learning techniques to estimate game model parameters from behavioral data.

Our computational models build on top of *interdependent security (IDS) games*, a model introduced by economists and risk-assessment experts Kunreuther and Heal to study investment decisions of strategic agents when facing direct and transfer risk exposure from other agents in the system. We first introduce *generalized IDS (α-IDS) games*, a model that extends IDS games where full investment can reduce transfer risk. In particular, α is a vector of probabilities, one for each agent, specifying the probability that the transfer risk will not be protected by the investment. In other words, agent i's investment can reduce indirect risk by probability $(1 - \alpha_i)$. We then extend from α-IDS games and introduce *interdependent defense (IDD) games*, a computational-game-theoretic framework for settings of interdependent security to study scenarios of multiple-defenders vs. a single-attacker in a network. For the variants of defense games we introduced, we study some computational aspects of computing Nash equilibria in those games.

Finally, we investigate the problem of *learning the games* form observed behavioral data. For this problem, we introduce a machine-learning generative model to learn the parameters of the games. As an application, we apply the learning model and use machine-learning techniques to estimate the parameters and structure of α-IDS games using the vaccination data from the Centers for Disease Control and Prevention (CDC) in the United States.

Dedication Page

For my family,

colleagues,

advisors, and

friends.

Contents

1 Introduction **1**

1.1 Components of My Thesis 2

1.1.1 Thesis Organization 2

1.2 Background: Game Theory Basic 4

2 Interdependent Security Games **10**

2.1 Preliminary: Parameters of IDS Games 11

2.2 Nash Equilibrium in IDS Games 14

3 α-IDS Games **17**

3.1 An Illustrative Example 17

3.2 The Model: α-IDS Games 19

3.3 Computing PSNE of α-IDS Games 21

3.3.1 SC α-IDS games 26

3.3.2 SS α-IDS games 27

3.4 Computing all NE in α-IDS games 27

3.4.1 Uniform-transfer SC α-IDS games 28

3.4.2 Uniform-transfer SS α-IDS games 32

3.4.3 Uniform-transfer SC+SS α-IDS games 33

3.4.4 Computing all MSNE of arbitrary α-IDS games is intractable, in general 35

3.5 Preliminary Experimental Results 42

3.5.1 Structure of the Graphs 42

3.5.2 Generating Parameters of α-IDS Games 43

3.5.3 Computing Approximate NE of α-IDS Games 43

3.6 Conclusion 46

4 Interdependent Defense Games 47

4.1 Mixed Strategies in IDD Games 49

4.2 Model Assumptions 50

4.3 PSNE of IDD Games 51

4.4 MSNE of IDD Games 53

4.4.1 Characterization 54

4.4.2 Security investment characteristics of MSNE 65

4.4.3 Relation to network structure 66

4.4.4 Computing All MSNE Efficiently 68

4.5 Experiments: Computing an ϵ-MSNE using BRGD 71

4.5.1 Computing an ϵ-MSNE using BRGD 73

4.5.2 Case Study: A randomly-generated instances of an Internet games at 0.005-MSNE 80

4.6 Conclusion and Open Problem 80

5 ϵ-MSNE of IDD Games 85

5.1 On the Complexity of Computing an MSNE 85

5.2 FPTAS to Compute ϵ-MSNE of Tree-like IDD Games 90

5.2.1 Directed Stars 91

5.2.2 Directed Trees 94

5.3 A Heuristic to Compute ϵ-MSNE 96

5.3.1 Evaluation of Heuristic on Internet Games 99

5.4 Conclusion 103

6 Learning from Mixed-Strategies 104

6.1 Introduction 104

6.2 A Framework to Learn Games from Data 107

6.2.1 A Generative Model for Behavioral Data on Joint-mixed-strategies 108

6.2.2 Learning Parameters of the Games via MLE 113

6.3 Application: Learning α-IDS Games 117

6.3.1 Generalized Interdependent Security Games 118

6.3.2 Learning the Structure and Parameters of α-IDS Games119

6.4 Preliminary Experiment 123

6.4.1 Parameters of the Players of Learned α-IDS game . . . 125

6.5 Conclusion 132

7 Conclusion 134

List of Figures

3.1 α-IDS Game of Zachary Karate Club at a NE. 19

3.2 3-SAT-induced SC+SS α-IDS game-graph. 22

3.3 Monotone 1 In 3-SAT-induced SC α-IDS game-graph 35

4.1 Histograms of Indegree and Outdegree of the Nodes of the Autonomous Systems from DIMES 72

4.2 Indegree and Outdegree of the Nodes of the Autonomous Systems from DIMES 72

4.3 Convergence Rate of Learning Dynamics 74

4.4 Attacker's Equilibrium Strategy on an Internet Game Instance (Fixed) 75

4.5 Attacker's Equilibrium Strategy and the Degrees of the Nodes 76

4.6 Attacker's and Site's ϵ-MSNE Strategies for a Randomly-Generated Internet Game 77

4.7 Attacker's Strategy at ϵ-MSNE 78

4.8 Attacker's ϵ-MSNE Strategy vs. Node Degrees. 79

4.9 The Structure of an Attack to the Internet 81

4.10 Attacker's Equilibrium Strategy vs Degrees of the Nodes at 0.05-MSNE 82

4.11 The degrees and strategies of the non-targeted nodes 83

5.1 Attack Distribution of Internet Game 99

5.2 Properties of our heuristic 101

5.3 Combing BRGD and our heuristic 101

5.4 Degrees of the targeted nodes at ϵ-MSNE 102

6.1 Sampled proportional of ϵ-MSNE in random α-IDS games . . 120

6.2 Choropleth map of vaccination percentage 124

6.3 State vaccination percentage 124
6.4 Players' Types 127
6.5 Histograms of the Δ values 128
6.6 Values of the Safety Functions 129
6.7 Learned game-graph 131
6.8 Equilibrium of the Learned α-IDS Game 132

List of Tables

1.1 Prisoner's Dilemma Game 5

3.1 Exact number of nodes and edges for different real-world graphs 43
3.2 Level of Investment of SC+SS α-IDS Games at Nash Equilibrium 45

4.1 Internet Games' Model Parameters 73

5.1 Selection of the constant c for our heuristic 100

Preface

Game theory, a topic that has been studied extensively both analytically and computationally over the recent years, models the interactions between agents in an environment or a state of the world. By interactions, we mean that each agent strategically selects an action given the actions of other agents. For examples, one can use game theory to model the interactions between the kicker and goalie in a penalty kick or a simple game of rock-paper-scissors between two players. When each agent is "happy" and has no incentive to deviate to other actions given the actions of other agents, we say that an equilibrium or a stable outcome has been reached.

Roughly speaking, a GT model consists of a set of players, and for each player, a set of actions, or pure strategies, as well as a payoff function that quantifies the player's preferences as a function of joint-action of all players. The general goal is to find a strategy (*equilibrium*) for each player, such that each player maximizes his/her payoff conditioned on the joint-strategy of the other players. An equilibrium is achieved when, given the joint-action of other players, the player has no other strategy that would result in a higher payoff. We can allow each player to play *mixed strategies*; randomize (with respect to some probability distribution) over their pure strategies.

Due to an increase number of attacks by hackers and terrorists, there has been quite a bit of recent research activity in the general area of game-theoretic models for terrorism settings that aim to understand the behavior of the attackers and the attackers' targets. My thesis is centered on introducing, studying, and applying several game-theoretic models to security.

Acknowledgements

I would like to express my sincere gratitude to to those who have helped me to get this far. Finishing this thesis and completing my PhD would not be possible without their words of wisdom, encouragement, and support.

I would also like to thank the National Science Foundation (NSF) for awarding me the graduate research fellowship and providing the main support of my thesis work.

Chapter 1

Introduction

Attacks carried out by hackers and terrorists such as the 9/11 attacks, the 2006 transatlantic aircraft plot, the Northwest Airlines Flight 253 Detroit bombing attempts, and the Yemen bomb scare over the last few years have led to increased efforts by both government and the private sector to create and adopt mechanisms to prevent future attacks. This effort has yielded a more focused research attention to models, computational and otherwise, that facilitate and help to improve (homeland) security for both physical infrastructure and cyberspace. In particular, there has been quite a bit of recent research activity in the general area of game-theoretic models [von Neumann and Morgenstern, 1944] for terrorism settings (see, e.g., Bier and Azaiez [2009] and Cárceles-Poveda and Tauman [2011]) and network security [Roy et al., 2010].

For example, Lye and Wing [2002] look at the interactions between an attacker and the (system) administrator using a two-player stochastic game (See Raghaven et al. [1990] for a reference to stochastic games). Liu [2003] focus on understanding the attacker's intent, objectives, and strategies and derive a (two-player) game-theoretical model. Recent work by [Jain et al., 2011, Kiekintveld et al., 2009, Korzhyk et al., 2010, 2011a,b] uses a Stackelberg game model, a game model in which the defender (or leader) commits to a mixed strategy to allocate resources to defend a set of nodes in the network, and the follower (or attacker) optimally allocates resources to attack a set of "targets" in the network given the leader's commitment. For examples, one can view the resources as some security mechanisms such as the locations to install security checkpoints and the assignments of air marshals to various flights. Therefore allowing the government to take preventive measures.

1.1 Components of My Thesis

My thesis is centered on introducing and applying several game-theoretic models [von Neumann and Morgenstern, 1944] to security setting. More specifically, my thesis consists of the following research components: (1) designing increasingly more realistic variants of defense games; (2) studying computational questions in defense games such as equilibria computation, computational implications of equilibria characterizations[1]; (3) designing better, more efficient algorithms and effective heuristics for defense problems; and (4) applying and designing new machine learning techniques to estimate game model parameters from behavioral data.

1.1.1 Thesis Organization

Chapter 2 We will begin by providing background information on the original interdependent security (IDS) games and related work. This chapter will form the basis of our security game models.

Chapter 3 Then, we introduce and study an extension of the IDS games where we call the games generalized IDS games (α-IDS games). Like traditional IDS games introduced by Kunreuther and Heal [2003, 2004, 2007], α-IDS games model agents' voluntary investment decisions when facing potential direct risk and transfer-risk exposure from other agents. A distinct feature of α-IDS games, however, is that full investment can reduce transfer risk. In particular, α is a vector of probabilities, one for each agent, specifying the probability that the transfer risk will not be protected by the investment. In other words, agent i's investment can reduce indirect risk by probability $(1 - \alpha_i)$. As a result, depending on the transfer-risk reduction level, generalized IDS games may exhibit strategic complementarity (SC) or strategic substitutability (SS) [Topkis, 1979, Vives, 1990]. We consider three variants of generalized IDS games in which players exhibit only SC, only SS,

[1] To put our computational contributions in context, note that deciding whether a graphical game (with bounded neighbors) has a PSNE is, in general, NP-complete (see, e.g., Gottlob et al. [2005]). For normal-form games, computing an MSNE in them is PPAD-complete, even in two-player multi-action games (see, e.g., Chen et al. [2009] and Daskalakis et al. [2009]). Also, in normal-form games, computing *all* MSNE is rarely achieved and counting-related problems are often #P-complete (see, e.g., Conitzer and Sandholm [2008]). However, these results are based on the "hard" instances of games.

and both SC+SS. We show that determining whether there is a pure-strategy Nash equilibrium (PSNE) in SC+SS-type games is NP-complete, while computing a single PSNE in SC-type games takes worst-case polynomial time. As for the problem of computing all mixed-strategy Nash equilibria (MSNE) efficiently, we produce a partial characterization. Whenever each agent in the game is indiscriminate in terms of the transfer-risk exposure to the other agents, a case that Kearns and Ortiz originally studied in the context of traditional IDS games in their NIPS 2003 paper, we can compute all MSNE that satisfy some ordering constraints in polynomial time in all three game variants. Yet, there is a computational barrier in the general (transfer) case: we show that the computational problem is as hard as the Pure-Nash-Extension problem, also originally introduced by Kearns and Ortiz [2004], and that it is NP-complete for all three variants. Finally, we experimentally examine and discuss the practical impact that the additional protection from transfer risk allowed in generalized IDS games has on MSNE by solving several randomly-generated instances of SC+SS-type games with graph structures taken from several real-world datasets.

Chapter 4 Starting from α-IDS games, we formally define and study Interdependent Defense (IDD) games in depth. In this chapter, we present several results that *fully* characterize their NE (PSNE and MSNE), which includes a *polynomial-time* algorithm to compute *all* MSNE for an important subclass of IDD games in which there is only one attack, the defender nodes are *fully transfer-vulnerable* (i.e., investing in security does nothing to reduce their external/transfer risk) and transfers are *one-hop.*[2] Moreover, we provide experimental results from applying learning-in-games (or best-response-gradient dynamics) heuristics to compute approximate NE to both fixed and randomly-generated instances of IDD games, with at most one simultaneous attack and one-hop transfers, on a very large Internet AS graph ($\approx 27K$ nodes and $\approx 100K$ edges). We call this subclass of IDD games *Internet games.*

Chapter 5 We continue our study of IDD games, and we further examine the problem of computing a NE in IDD games. We show that an efficient algorithm to determine whether some attacker's strategy can be a part of

[2] We note that the original IDS games [Kunreuther and Heal 2003, 2004, 2007] are also *fully transfer-vulnerable* and assume one-hop transfers.

a NE in an instance of IDD games is unlikely to exist. Yet, we provide a dynamic programming algorithm to compute an approximate NE when the graph/network structure of the game is a directed tree with a single source, and show that it is a fully polynomial time approximation scheme (FPTAS). We also introduce an improved heuristic to compute an approximate NE on arbitrary graph structures. Our experiments show that our heuristic is more efficient, and provides better approximations, than best-response-gradient dynamics for the case of Internet games.

Chapter 6 We introduce a statistical framework to learn the structures and parameters of games given a set of possible (approximate) MSNE. Under our framework, we show that maximizing the log-likelihood of the game is equivalent to maximizing the number of (approximate) MSNE in the data under some mild assumptions. Moreover, using the vaccination data from the Center for Disease Control and Prevention (CDC) in the United States, we demonstrate the effectiveness of our framework on learning the parameters of the α-IDS games that best fit the data. Although our focus in this chapter is to learn α-IDS games, we can also apply it to learn IDD games. Due to dataset availability, we settle for learning α-IDS games and leave learning IDD games as a future exploration, conditional on the available of attacker-defender security data.

1.2 Background: Game Theory Basic

Game theory is the study of interaction among *independent* and *self-interested* agents or players. The notion of independent means that the basic modeling unit is the individual. The modeling unit includes the agent's *beliefs*, *preferences*, and *actions*. In our work, we are only interested in noncooperative game theory as opposed to cooperative game theory, where there are binding agreements among some agents. Moreover, self-interested does not mean that the agents want to cause harm to each other, or only care about themselves. Instead, they have their own description of which states of the world they like and that they act trying to bring about these states.

We use a *utility function* to model an agent's degree of interest. Roughly speaking, a utility function is a mapping from states of the world to real numbers. These numbers are interpreted as measures of an agent's level of "happiness" in the given states. When the agent is uncertain about which state of the world he faces, his utility is defined as the expected value of his utility function with respect to the appropriate probability distribution. Each agent's goal is to choose the set of actions (or probability distribution over the set of actions) that maximizes (*expected*) utility.

Model Interaction. The normal-form representation (or normal-form game) is arguably the most fundamental representation in game theory to model interaction among agents.

Definition 1. *A* (finite, n-person) normal-form game *is a tuple* (N, A, u) *where*

- N *is a finite set of* n players*, indexed by* i*;*
- $A = A_1 \times ... \times A_n$*, where* A_i *is a finite set of actions or pure strategies available to player* i*, and each vector* $a = (a_1, ..., a_n) \in A$ *is called an* action profile*;*
- $u = (u_1, ..., u_n)$ *where* $u_i : A \rightarrow R$ *is a real-valued* utility (or payoff) function *for player* i*.*

A natural way to represent normal-form games is via an n-dimensional matrix. Below is a classic example of a 2-person prisoner's dilemma game in normal form [Fudenberg and Tirole, 1991]. Each row corresponds to a possible action for player 1, each column corresponds to a possible action for player 2, and each cell corresponds to one possible outcome. Each player's utility for an outcome is written in the cell corresponding to that outcome, with player 1's utility listed first.

	Cooperate	Defect
Cooperate	-1, -1	-4, 0
Defect	0, -4	-3, -3

Table 1.1: Prisoner's Dilemma Game

Strategies in Normal-Form Game. There are many strategies available to a player. An example of a simple strategy for a player is for that player to play one action (or pure-strategy). A player can choose a more complex strategy in which he plays the actions according to some probability distribution. This such strategy is also known as a mixed strategy.

Let (N, A, u) be a normal-form game. For any set D, let $\Delta(D)$ be the set of all probability distributions over D.

Definition 2. *The set of mixed strategies of player i is $X_i = \Delta(A_i)$. The set of pure strategies for player i is $X_i = A_i$.*

The set of pure-strategy and mixed-strategy profiles is simply the Cartesian product of the individual pure-strategy set $(A_1 \times ... \times A_n)$ and mixed-strategy set $(X_1 \times ... \times X_n)$.

The following definition gives us a way to compute the expected utility value for each player under mixed-strategy profile.

Definition 3. *Given a normal-form game (N, A, u) and the mixed-strategy profile $x = (x_1, ..., x_n)$, player i's* expected utility *u_i is*

$$u_i(x) \equiv \textstyle\sum_{a \in A} u_i(a) \prod_{j=1}^{n} x_j(a_j),$$

where $x_j(a_j)$ is the probability of player j plays the action a_j according to x_j.

Note that pure-strategy profile is equivalent to a mixed-strategy profile where the probability of playing an action is one.

Equilibrium (Solution) Concept. Given a game, each player strategically selects a strategy given the strategies of other players. In particular, when each player is "happy" and has no incentive to deviate to other strategies given the strategies of other players, we say that an equilibrium or a stable outcome has been reached.

There are many equilibrium concepts (see Fudenberg and Tirole [1991] for other concepts). One that has been studied extensively, both analytically and computationally, over the recent years is the concept of *Nash equilibrium* (NE). Intuitively, a NE is a stable strategy profile: no player would want to change his strategy if he knew what strategies the other players were following.

Definition 4. *A strategy profile $x^* = (x_1^*, ..., x_n^*)$ is a NE if, for all players i and for all strategies $x_i' \in X_i$, $u_i(x_i^*, x_{-i}^*) \geq u_i(x_i', x_{-i}^*)$.*

Alternatively, we can define NE in terms of best response.

Definition 5. *Player i's* best response *to the strategy profile x_{-i} is a mixed strategy $x_i^* \in X_i$ such that $u_i(x_i^*, x_{-i}) \geq u_i(x_i, x_{-i})$ for all strategies.*

Definition 6. *A strategy profile $x^* = (x_1^*, ..., x_n^*)$ is a NE if, for all players i, x_i^* is a best response to x_{-i}^*.*

In fact, there are alternative mathematical expressions for NE that are more computationally amenable.

Proposition 1. *A strategy profile $x^* = (x_1^*, ..., x_n^*)$ is a NE if, for all players i, $u_i(x_i^*, x_{-i}^*) \geq u_i(a_i, x_{-i}^*)$ for all $a_i \in A_i$.*

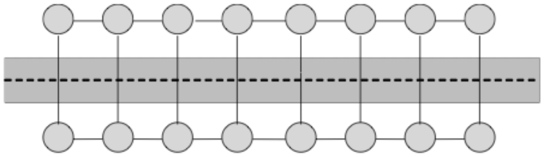
Proof. Suppose that have a strategy profile $x^* = (x_1^*, ..., x_n^*)$ such that for all players i, $u_i(x_i^*, x_{-i}^*) \geq u_i(a_i, x_{-i}^*)$ for all $a_i \in A_i$. For all players i, let $a_i^{max} \in A_i$ such that $a_i^{max} \in \arg\max_{a_i \in A_i} u_i(a_i, x_{-i}^*)$. For any $x_i' \in X_i$,

$$\begin{aligned} u_i(x^*) &\geq \sum_{a_i \in A_i} x_i'(a_i) u_i(a_i^{max}, x_{-i}^*) \geq \sum_{a_i \in A_i} x_i'(a_i) u_i(a_i, x_{-i}^*) \\ &= \sum_{a_i \in A_i} x_i'(a_i) \sum_{a_{-i} \in A_{-i}} u_i(a_i, a_{-i}) \prod_{j=1, j \neq i}^{n} x_j^*(a_j) = u_i(x_i', x_{-i}^*) \end{aligned}$$

where the first inequality is because $\sum_{a_i \in A_i} x_i'(a_i) = 1$ and the second inequality is because $a_i^{max} \in \arg\max_{a_i \in A_i} u_i(a_i, x_{-i}^*)$. Thus, x^* is a NE. □

Throughout the text, we will use the short-hand expressions PSNE and MSNE to distinguish the set of NE resulting from all the players playing pure-strategies and the players playing mixed-strategies, respectively. However, a PSNE is also an MSNE and the general term NE can be interpreted broadly as MSNE. In either case, the definitions and proposition above hold for both PSNE and MSNE.

A more compact representation: Graphical Games. In a more practical sense, the normal-form representation of games is too large and unrealistic for real life when the number of players is large. We can use graphical games to represent normal-form games compactly. Graphical games use graphical models to capture the payoff (conditional) independence structure of the games. A classic example is Road games (see below) [Shoham and Leyton-Brown, 2009].



Consider n players, each of whom has a piece of land alongside a road. Each agent has to choose what to build on his/her land. In this example, graph encodes the modeling assumption. His/Her payoff depends on what his neighbors (the connected nodes) have built.

Definition 7. *Let $G = (N, E)$ be the graph defined on a set of nodes N and and a set of edges E. For every $i \in N$, let $n(i) = \{i\} \cup \{j \mid (j, i) \in E\}$ be the set of neighboring nodes of i and i.*

Definition 8. *A graphical game is a tuple $(G = (N, E), A, u)$, where:*

- *$G = (N, E)$ is a set of n vertices, representing players, and E is a set of undirected edges connecting the nodes N;*
- *$A = A_1 \times ... \times A_n$, where A_i is the set of actions available to agent i; and*
- *$u = (u_1, ..., u_n)$ where $u_i : A_{n(i)} \to R$, where $A_{n(i)} = \prod_{j \in n(i)} A_j$.*

Note that an edge between two vertices in the graph can be view as the two players are able to affect each other's payoffs. Moreover, the graphical structure can also be directed and the definition can be modified easily by simply replacing $n(i)$ by $\mathrm{Pa}(i) \cup \{i\}$ for all players $i \in N$ where $\mathrm{Pa}(i)$ is the parent of i in the directed graph representation.

Computing Nash Equilibrium. The complexity of computing a PSNE depends on the representation of the games. For instance, computing a PSNE in the classic normal-form representation where the utilities (for each possible combination of actions) are given via a (large) table/matrix can be done in logarithmic space and in polynomial time [Gottlob et al., 2005]. However, the complexity of computing a PSNE is rather negative as shown by Gottlob et al. [2005].

Theorem 1. *Determining whether a graphical games of bounded neighbors has a PSNE is NP-complete.*

However, if the graph (in graphical games) has constant treewidth and small neighborhood, computing PSNE can be solved in polynomial time by formulating the computational problem as a constraint satisfaction problem [Gottlob et al., 2005]. Therefore, allowing us to apply the standard CSP algorithms [Dechter, 2003] to find a PSNE if it exists.

While the hardness of computing a PSNE depends on the representation of the games. The following results hold in general, regardless of representations, when computing MSNE. Every game has a MSNE Nash [1950] while Chen et al. [2009] and Daskalakis and Papadimitriou [2005], Daskalakis et al. [2009], Chen and Deng [2006] show that computing a MSNE is PPAD-complete even in two-player multi-action games.

Theorem 2. *[Nash, 1950] Every game with a finite number of players and action profiles has at least one MSNE.*

Theorem 3. *[Chen and Deng, 2006] [Daskalakis and Papadimitriou, 2005], [Daskalakis et al., 2009] The problem of finding a MSNE of a general sum finite game with two or more players is PPAD-complete.*

Chapter 2

Interdependent Security Games

Interdependent security (IDS) games are one of the earliest models resulting from a game-theoretic approach to model security in non-cooperative environments composed of free-will self-interested individual decision-makers. Originally introduced and studied by economists Kunreuther and Heal [2003, 2004, 2007], IDS games model the general abstract security problems in which an individual within a population considers whether to voluntarily invest in some protection mechanisms or security against a risk they may face, knowing that the cost-effectiveness of the decision depends on the investment decisions of others in the population because of transfer risks (i.e., the "bad event" may be transferable from a compromised individual to another).

In their work, Kunreuther and Heal [2003, 2004, 2007] provided several examples based on their economics, finance, and risk management expertise. As a canonical example of the real-world relevance of IDS settings and the applicability of IDS games, Heal and Kunreuther [2005a] used this model to describe problems such as airline baggage security. In their setting, individual airlines may choose to invest in additional complementary equipment to screen passengers' bags and check for hazards such as bombs that could cause damage to their passengers, planes, buildings, or even reputations. One can also view bomb packages as passengers where they often only get screened at their initial airport, resulting in decisions where an attacker boards a plane in, for instance, an African airport to connect to a Europe-US flight [O'Connor and Schmitt, 2009]. However, mainly due to the large amount of traffic volume, it is impractical for an airline to go beyond applying security checks to bags incoming from passengers and include checks to baggage or cargo transferred from other airlines. On the other hand, if an airline invests in security, they can still experience a bad event if the bag was transferred from an airline that does not screen incoming bags, rendering their investment useless. [1] Thus, we can see how the cost-effectiveness of an investment can be highly dependent on others' investment decisions. Another recent application of the IDS model is on container shipping transportation [Gkonis and Psaraftis, 2010]. They use the IDS model to study the effect of investment decision on container screening of ports have on their neighboring ports. Furthermore, IDS games can be used to model other practical real-world situations such as vaccination Heal and Kunreuther [2005b]. See Laszka et al. [2014] for a survey on IDS games and other variants of IDS games.

2.1 Preliminary: Parameters of IDS Games

We start by looking at *Interdependent Security (IDS) games* and define the parameters and rules governing this model. There are n players in the IDS games and the players are indexed by $[n] = \{1, 2, ..., n\}$. Each player $i \in [n]$ has a *choice* to *invest* or *not invest*. Thus, the action set of a player i is denoted by the set $\{0, 1\}$ and, for convenient, we let $a_i = 1$ denote the action of invest and let $a_i = 0$ denote the action of not invest.

For each player $i \in [n]$, let $C_i \in \mathbb{R}^+$ be the *cost of investment* and $L_i \in \mathbb{R}^+$ be the *loss induced by the bad event*.

For each player $i \in [n]$, we let $p_i \in [0, 1]$ to be the probability that player i will experience a bad event because of a direct contamination. We will refer the parameter p_i as *direct risk* or *internal risk* of player i. The standard IDS model assumes that investing will completely protect the player from direct contamination; hence, internal risk is only possible when $a_i = 0$.

For each player $i \in [n]$ and for each player $j \neq i$, the *indirect risk* or *transfer risk* is denoted by $q_{ji} \in [0, 1]$. The transfer risk is the probability that player j is directly "contaminated," does not experience the bad event but transfers it to player i who ends up experiencing the bad event. The player receiving the transfer still has the chance of not experiencing the bad event. The IDS model also assumes that the interactions between players are unaffected by investment, so regardless of one's investment, one's transferred risk is the same.

[1] Note that even if full screening were performed, the Christmas Day 2009 episode in Detroit [O'Connor and Schmitt, 2009] serves as a reminder that transfer risk still exists.

As pointed out by Kearns and Ortiz [2004], the original IDS model imposes an implicit global constraint on the internal and transfer risk parameters such that $p_i + \sum_{j \in \text{Pa}(i)} q_{ji} \leq 1$ for all $i \in [n]$.

Denote by $a \equiv (a_1, \ldots, a_n) \in \{0,1\}^n$ the *joint-action* or *pure-strategy* profile of all n players. Also denote by a_{-i} the joint-action or action profile of all players except i and for any subset $I \subseteq [n]$ of players, denote by a_I the sub-component of the joint-action corresponding to those players in I only.

We now formally define a (directed) graphical game [Kearns et al., 2001, Kearns, 2007] version of IDS games, as first introduced by Kearns and Ortiz [2004]. The graphical structure is induced by the transfer risk parameters. In particular, the parameters q_{ij}'s induce a directed graph $G = ([n], E)$ such that $E \equiv \{(i,j) \text{ for } i,j \in [n] \mid q_{ij} > 0\}$. Let $\text{Pa}(i) \equiv \{j \in [n] \mid q_{ji} > 0\}$ be the set of players that are *parents* of player i in G (i.e., the set of players that player i is exposed to via transfers), and by $\text{PF}(i) \equiv \text{Pa}(i) \cup \{i\}$ the *parent family* of player i, which *includes* i. Denote by $k_i \equiv |\text{PF}(i)|$ the size of the parent family of player i. Similarly, let $\text{Ch}(i) \equiv \{j \in [n] \mid q_{ij} > 0\}$ be the set of players that are *children* of player i (i.e., the set of players to whom player i can present a risk via transfer) and $\text{CF}(i) \equiv \text{Ch}(i) \cup \{i\}$ the *(children) family* of player i, which *includes* i. Note that the above definitions and notations are standard in graph theory for directed graphs.

Giving the graph that is induced by the transfer risks, one important notion in IDS games is the *safety* of the players. Indeed, the safety of a player i depends not only on the transfer risks but also on the investment decisions of other players that could transfer the bad event to i. To compute the safety of player i, we first compute the the *probability that player* i *is safe from player* j. More specifically, the probability that player i is safe from player j, as a function of player j's decision, is

$$e_{ij}(a_j) \equiv a_j + (1 - a_j)(1 - q_{ji}) = (1 - q_{ji})^{1-a_j},$$

because, in the standard IDS games of Kunreuther and Heal [2003, 2004, 2007], if j invests, then it is impossible for j to transfer the bad event, while if j does not invest, then j either experiences the bad event or transfers it to another player, but never both.

Because the transfer risks to i and the safety functions of i from other players are independent, i's *overall safety* from *all* other players is

$$s_i(a_{\text{Pa}(i)}) \equiv \prod_{j \in \text{Pa}(i)} e_{ij}(a_j),$$

and equivalently his *overall risk* from other players is

$$r_i(a_{\mathrm{Pa}(i)}) \equiv 1 - s_i(a_{\mathrm{Pa}(i)}).$$

Note that each player's overall safety (and risk) is a direct function of its parents only, *not* all other players. Moreover, it is important to realize that, in the IDS model, there is only one bad event in the system and the bad event such as bomb explosion can only occur once [Heal and Kunreuther, 2003]. Therefore, the above risk function captures that the probabilities that the bad event will be transferred from the parents that do not invest in security.

From these definitions, we obtain player i's overall cost, the cost of joint-action $a \in \{0,1\}^n$, corresponding to the (binary) investment decision of all players, is

$$\begin{aligned} M_i(a_i, a_{\mathrm{Pa}(i)}) \equiv & a_i[C_i + r_i(a_{\mathrm{Pa}(i)})L_i] + \\ & (1-a_i)[p_i + (1-p_i)r_i(a_{\mathrm{Pa}(i)})]L_i \ . \end{aligned}$$

Given the cost function, the goal of the player is to select the action that minimizes its cost given the actions of others. Whether players invest depends solely on what they can gain or lose by investing. If the overall cost of investing is less than the overall cost of not investing, the player will invest. Applying this logic to cost function M_i, player i will invest if

$$C_i + r_i L_i < [p_i + (1-p_i)r_i]L_i$$

so that the investment cost and the losses due to a transferred event do not outweigh the losses from an internal or transferred bad event. Similarly, if the inequality in the last expression is reversed or is replaced by equality, player i will not invest or would be indifferent, respectively.

Rearranging the expression for the best-response conditions given in the last equation and letting

$$\Delta_i \equiv \frac{C_i}{p_i L_i},$$

the *cost-to-expected-loss ratio* of player i, we get the following *best-response correspondence* $\mathcal{BR}_i : \{0,1\}^{k_i - 1} \to 2^{\{0,1\}}$ for player i:

$$\mathcal{BR}_i(a_{\mathrm{Pa}(i)}) \equiv \begin{cases} \{1\}, & \text{if } s_i(a_{\mathrm{Pa}(i)}) > \Delta_i, \\ \{0\}, & \text{if } s_i(a_{\mathrm{Pa}(i)}) < \Delta_i, \\ \{0,1\}, & \text{if } s_i(a_{\mathrm{Pa}(i)}) = \Delta_i. \end{cases} \qquad (2.1)$$

In other words, whether it is cost-effective for player i to invest or not depends on a simple threshold condition on his safety. For the original IDS model, the question a player faces is, does he feel safe enough from others?

Definition 9. *A pure-strategy profile $a^* \in \{0,1\}^n$ is a PSNE of an IDS game if $a_i^* \in \mathcal{BR}_i(a^*_{Pa(i)})$ for all players $i \in [n]$ (i.e., a^* is a mutual best-response).*

The cost function can be easily extended to accommodate the mixed-strategy setting in which each player plays the actions with some probability. Formally, for each player i, we let $\mathcal{A}_i$ to be a Bernoulli random variable that can take a value of zero or one. We denote $x_i = Pr(\mathcal{A}_i = 1)$ to be the probability that $\mathcal{A}_i$ is equal to one. In other words, x_i denotes the mixed strategy of player i that specifies i's probability of playing the action invest. Thus, player i's *expected cost function* is the expectation over the cost function when the pure actions are represented by the Bernoulli random variables $\mathcal{A}_i$'s. With a slight abuse of notation, player i's expected cost function is

$$
\begin{aligned}
&M_i(x_i, x_{\mathrm{Pa}(i)}) \equiv \mathbb{E}[M_i(\mathcal{A}_i, \mathcal{A}_{\mathrm{Pa}(i)})] \\
&= \mathbb{E}[\mathcal{A}_i[C_i + r_i(\mathcal{A}_{\mathrm{Pa}(i)})L_i] + (1-\mathcal{A}_i)[p_i + (1-p_i)r_i(\mathcal{A}_{\mathrm{Pa}(i)})]L_i] \\
&= x_iC_i + \mathbb{E}[\mathcal{A}_i r_i(\mathcal{A}_{\mathrm{Pa}(i)})]L_i + (1-x_i)p_iL_i + \mathbb{E}[(1-\mathcal{A}_i)(1-p_i)r_i(\mathcal{A}_{\mathrm{Pa}(i)})]L_i \\
&= x_iC_i + x_ir_i(x_{\mathrm{Pa}(i)})]L_i + (1-x_i)p_iL_i + (1-x_i)(1-p_i)r_i(x_{\mathrm{Pa}(i)})L_i \\
&= x_i[C_i + r_i(x_{\mathrm{Pa}(i)})L_i] + (1-x_i)[p_i + (1-p_i)r_i(x_{\mathrm{Pa}(i)})]L_i \ ,
\end{aligned}
$$

where the second equality is by linearity of expectation and the third equality is by the expected value of mutually independent random variables. Moreover, the best-response correspondence of each player i can be redefined in terms of mixed strategy where $\mathcal{BR}_i : [0,1]^{k_i-1} \to S \subseteq [0,1]$.

Definition 10. *A mixed-strategy profile $x^* \in [0,1]^n$ is an MSNE of an IDS game if $x_i^* \in \mathcal{BR}_i(x^*_{Pa(i)})$ for all players $i \in [n]$.*

We will conclude this chapter by discussing the results on computing NE in IDS games.

2.2 Nash Equilibrium in IDS Games

One central question in studying game-theoretical model is the characterization and computation of NE. Below, we present previous results on finding NE in IDS games.

In regard to the existence of a PSNE, Heal and Kunreuther [2004] and Kearns and Ortiz [2004] show that there always exists one.

Theorem 4. *There exists a PSNE for any n-player IDS game. In addition, a PSNE in time $O(n^2)$.*

The key to show the above theorem is based on the following constructive proof and algorithm. The idea is to start with all players playing the action not invest. Given that, we check to see whether each player is playing a best-response according to Equation 2.1. If all of the players are playing according to their best-response, then we have a PSNE and we are done. Otherwise, there must be some players that are "unhappy" playing the action invest. Then, we switch the "unhappy" players to play the action not invest and keep the other players' action to invest. Once again, with those players playing invest and the remaining players playing not invest, we check to see if every player is playing a best response. If so, we are done. Otherwise, there must be some other players that are not playing their best-responses. We change their actions to invest and we repeat until every player is playing best response to each other (in worst case, everybody invests) [2].

Given the existence of a PSNE in IDS games, Heal and Kunreuther [2004] and Kearns and Ortiz [2004] investigate the question of finding *all NE* in uniform-transfer (indiscriminate) IDS games analytically and computationally, respectively. A uniform-transfer IDS game is an IDS game in which the transfer probability of a node is independent of the destination, that is, for all $i \neq j$, $q_{ij} = \delta_j$ for some value $\delta_j \in (0, 1]$. Kearns and Ortiz [2004] show that there exists a polynomial algorithm to compute all NE in uniform-transfer IDS games.

However, the result is quite negative for finding all NE in the non-uniform case. In particular, Kearns and Ortiz [2004] study a problem in which they call Pure-Nash Extension problem. The Pure-Nash Extension problem for any n-player game with binary actions takes a description of the game and a partial assignment $a \in \{0, 1, *\}^n$ as inputs and output a complete assignment that agrees with a (a PSNE) or "none".

Even in a slightly more general case of IDS games where all $i \neq j$, $q_{ij} \in \{\delta, 0\}$ for $\delta \in (0, 1]$ are allowed to have only two values and $|\mathrm{Ch}(i)| \leq 3$ for

[2] The same reasoning work starting with everyone playing the action invest and asking whether there are "unhappy" players (i.e., playing invest is never a good strategy regardless of what other players are playing) and recursively applying the same argument.

all $i \in [n]$, the Pure-Nash Extension problem for this version of IDS games is NP-complete. Therefore, the existence of an efficient algorithm to find all NE in general IDS games is unlikely.

Chapter 3

Generalized Interdependent Security Games[1]

In the standard IDS game model of Kunreuther and Heal [2003, 2004, 2007], investment in security does not reduce transfer risks. However, in some IDS settings (e.g., vaccination and cyber-security), it is reasonable to expect that security investments would include mechanisms to reduce transfer risks. This motivates our first modification to traditional IDS games: allowing the investment in protection to not only make us safe from direct attack but also partially reduce (or even eliminate) the transfer risk. Let us start off with a motivating example.

3.1 An Illustrative Example

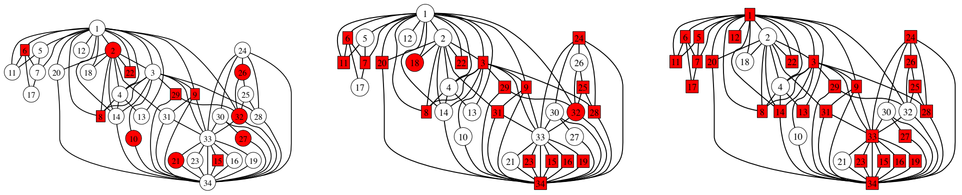
Let us be more concrete and consider an application of IDS games from Heal and Kunreuther [2004], Kearns and Ortiz [2004]. Imagine that you are an owner of an apartment. One day, there was a fire alarm in the apartment complex. Luckily, it was nothing major: nobody got hurt. As a result, you realize that your apartment can be easily burnt down because you do not have any fire extinguishing mechanism such as a sprinkler system. However, as you wonder about the cost and the effectiveness of the fire extinguishing mechanism, you notice that the fire extinguishing mechanism can only protect your apartment if a small fire originates in your apartment. If a fire

[1] A part of this chapter has appeared in the proceedings of the *Twenty-eighth Annual Conference on Neural Information Processing Systems (NIPS 2014).*

originates in the floor below, or above, or even the apartment adjacent to yours, then you are out of luck: by the time the fire gets to your apartment, the fire would be fierce enough already. You realize that if other apartment owners invest in fire extinguishing mechanisms, the likelihood of their fires reaching you decreases drastically. As a result, you debate whether or not to invest in the fire extinguishing mechanism given whether or not the other owners invest in fire extinguishing mechanisms. Indeed, making things more interesting, you are not the only one going through this decision process; assuming that everybody is concerned about their safety in the apartment complex, everybody in the apartment complex wants to decide whether or not to invest in fire extinguishing mechanisms given the individual decision of other owners.

To be more specific, as an IDS game, the players are the apartment owners, each apartment owner needs to make a decision as to whether or not to invest in the fire extinguishing mechanism based on cost, potential loss, as well as the direct and indirect (transfer) risks. The direct risk here is the chance that a player will start a fire (e.g., forgetting to turn off gas burners or overloading electrical outlets) and potentially burn his apartment down. The transfer risk here is the chance that a fire from somebody else's (unprotected) apartment will spread to other apartments. Moreover, as a feature of the IDS games, transfer risk comes from the direct neighbors conditional and cannot be re-transferred. In other words, the player only concerns about the risks from the direct neighbors. From the perspective of the player, the player does not care about what happens to him after the "bad event" (i.e., burning). However, the player does indirectly care about the transfer risks but conditionally on the actions of other players that can transfer to him directly. As we saw earlier, the player's cost function is directly independent from other players in the system.

Note that in the apartment complex example, the fire extinguishing mechanism does not protect an agent from fires that originate from other apartments. *In this work, we consider a more general, and possibly also more realistic, framework of IDS games where investment can partially protect the indirect risk* (i.e., investment in the fire extinguishing mechanism can partially extinguish some fires that originate from others). To distinguish the naming scheme, we will call these *generalized IDS games* as *α-IDS games* where α is a vector of probabilities, one for each agent, specifying the probability that the transfer risk will *not* be protected by the investment. In other words, agent i's investment can reduce indirect risk by probability (1-α_i).



$\alpha \sim \mathcal{N}(0.4, 0.2)$
(a)

$\alpha \sim \mathcal{N}(0.6, 0.2)$
(b)

$\alpha \sim \mathcal{N}(0.8, 0.2)$
(c)

Figure 3.1: **α-IDS Game of Zachary Karate Club at a NE.** Legend: Square $\equiv$ SC player, Circle $\equiv$ SS player, Shaded $\equiv$ Invest, and Not Shaded $\equiv$ No Invest, $\mathcal{N}(m, v) = Normal(mean, variance)$

Given an α, the players can be partitioned into two types: the SC type and the SS type. The *SC players* behave *strategic complementarily*: they invest if sufficiently many people invest. On the other hand, the *SS players* behave *strategic substitutability*: they do not invest if too many people invest.

As a preview of how the α can affect the number of SC and SS players and NE, which is the solution concept used here (formally defined for the α-IDS games in the next section), Figure 5.1 presents the result of our simulation of an instance of SC+SS α-IDS games using the Zachary Karate Club network [Zachary, 1977] where α is generated independently according to normal distribution with mean (a) 0.4, (b) 0.6, and (c) 0.8 and a common standard deviation of 0.2 for each player. The nodes are the players, and the edge between nodes u and v represents the potential transfers from u to v and v to u. The SC players are denoted by squares and the SS players are denoted by circles. Those that make an investment in an NE are shaded. As we increase α's value, the number of SC players increases while the number of SS players decreases. Interestingly, almost all of the SC players invest, and all of the SS players are "free riding" as they do not invest at the NE.

3.2 The Model: α-IDS Games

We incorporate this factor by introducing a new real-valued parameter $\alpha_i \in [0, 1]$ representing the probability that a transfer of a potentially bad event will go *unblocked* by i's security, assuming i has invested. Said differently, the parameter α_i models the degree to which investment in security can potentially reduce player i's transfer risk. Thus, we redefine player *i's* overall cost as [2]

$$\begin{aligned} M_i(a_i, a_{\mathrm{Pa}(i)}) \equiv & a_i[C_i + \alpha_i r_i(a_{\mathrm{Pa}(i)})L_i] + \\ & (1 - a_i)[p_i + (1 - p_i)r_i(a_{\mathrm{Pa}(i)})]L_i \ . \end{aligned}$$

Note that the safety function describes the situation where a player j can only be "risky" to player i if and only if j does not invest in protection.

While a *syntactically* minor addition to the traditional IDS model, the parameter α introduces a major *semantic* difference and an additional complexity over the traditional model. The semantic difference is perhaps clearer from examining the best response of the players: player i invests if

$$\begin{aligned} & C_i + \alpha_i r_i(a_{\mathrm{Pa}(i)})L_i < [p_i + (1 - p_i)r_i(a_{\mathrm{Pa}(i)})]L_i \\ \Leftrightarrow & \frac{C_i}{L_i} < p_i + (1 - p_i)r_i(a_{-i}) - \alpha_i r_i(a_{-i}) \\ \Leftrightarrow & R_i - p_i < (1 - p_i - \alpha_i)r_i(a_{\mathrm{Pa}(i)}) \ . \end{aligned}$$

The expression $(1 - p_i - \alpha_i)$ is positive when $\alpha_i < 1 - p_i$ and negative when $\alpha_i > 1 - p_i$. The best-response condition flips when the expression is negative. When $\alpha_i = 1 - p_i$, player i's investment decision simplifies because the player's internal risk fully determines the optimal choice. In other words, the best-response of player i is independent of other players' actions; player i behaves as a disconnected node in the game graph.

In fact, the parameter α induces a partition of the set of players based on whether the corresponding α_i value is higher or lower than $1 - p_i$. We will call the set of players with $\alpha_i > 1 - p_i$ the set of *strategic complementarity (SC)* players. SC players exhibit as optimal behavior that their preference for investing *increases* as more players invest: they are "followers." The set of players with $\alpha_i < 1 - p_i$ is the set of *strategic substitutability (SS)* players. In this case, SS players' preference for investing *decreases* as more players invest: they are "free riders."

For all $i \in SC$, let $\Delta_i^{sc} \equiv 1 - \frac{R_i - p_i}{1 - p_i - \alpha_i}$; similarly for Δ_i^{ss}, for $i \in SS$. We can define the *best-response correspondence* $\mathcal{BR}_i^{sc} : \{0, 1\}^{k_i - 1} \rightarrow 2^{\{0,1\}}$ for

[2] A possible generalization, which we do not pursue here, may also consider α_i a function of $\mathrm{Pa}(i)$.

player $i \in SC$ as

$$\mathcal{BR}_i^{sc}(a_{\text{Pa}(i)}) \equiv \begin{cases} \{0\}, & \Delta_i^{sc} > s_i(a_{\text{Pa}(i)}), \\ \{1\}, & \Delta_i^{sc} < s_i(a_{\text{Pa}(i)}), \\ \{0,1\}, & \Delta_i^{sc} = s_i(a_{\text{Pa}(i)})\ . \end{cases}$$

The *best-response correspondence* $\mathcal{BR}_i^{ss}$ *for player* $i \in SS$ is similar, except that we replace Δ_i^{sc} by Δ_i^{ss} and "reverse" the strict inequalities above. Therefore, the *best-response correspondence* $\mathcal{BR}_i^{ss} : \{0,1\}^{k_i-1} \to 2^{\{0,1\}}$ for player $i \in SS$ is

$$\mathcal{BR}_i^{ss}(a_{\text{Pa}(i)}) \equiv \begin{cases} \{0\}, & \Delta_i^{ss} < s_i(a_{\text{Pa}(i)}), \\ \{1\}, & \Delta_i^{ss} > s_i(a_{\text{Pa}(i)}), \\ \{0,1\}, & \Delta_i^{ss} = s_i(a_{\text{Pa}(i)})\ . \end{cases}$$

As before, we consider the mixed-strategy setting and denote x_i to be the probability that i plays the action invest. Player i's decision depends on *expected* cost, and, with abuse of notation, we denote it by $M_i(x)$.

Definition 11. *A pure-strategy profile $a \in \{0,1\}^n$ is a PSNE of an α-IDS game if (1) for each player $i \in SC$, $a_i \in \mathcal{BR}_i^{sc}(a_{Pa(i)})$ and (2) for each player $i \in SS$, $a_i \in \mathcal{BR}_i^{ss}(a_{Pa(i)})$. Replacing the pure-strategy profile a with a mixed-strategy profile $x \in [0,1]^n$ in the equilibrium condition and the respective functions it depends on, this leads to the condition for x being a MSNE.*

For convenient, in the following, we will denote the α-IDS games consist of only SC players, only SS players, and both SC and SS players as SC α-IDS games, SS α-IDS games, and SC+SS α-IDS games, respectively.

3.3 Computational results for Finding a PSNE in α-IDS games

In this section, we present and discuss the results of our computational study of α-IDS games. We begin by considering the problem of computing PSNE, then moving to the more general problem of computing MSNE.

In this subsection, we look at the complexity of determining a PSNE in α-IDS games, and determine if one exists. Our first result follows.

Theorem 5. *Determining whether there is a PSNE in n-player SC+SS α-IDS games is NP-complete.*

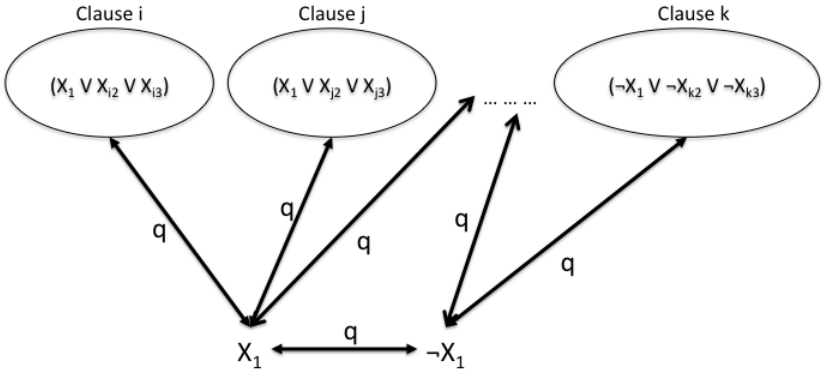


Figure 3.2: **3-SAT-induced SC+SS α-IDS game-graph.**

Proof. For formality, we first define the notations that will be used in the proof. In particular, we consider the problem of determining whether there is a PSNE in SC+SS α-IDS games and denote the instances with PSNE as

$$\begin{aligned}\alpha\text{-}IDS \;\; = \;\; \{ \; & \big([n], (C_i)_{i\in[n]}, (\alpha_i)_{i\in[n]}, (L_i)_{i\in[n]}, (p_i)_{i\in[n]}, (q_{ji})_{j,i\in[n],i\neq j}\big) : \\ & \text{there exists a PSNE in } \mathbb{G} \; \}.\end{aligned}$$

We will reduce our problem from a variation of 3-SAT (also called Boolean Satisfiability Problem) where each clause of the 3-SAT has exactly three variables and consists of either negated variables or (un-negated) variables. We use the term variable(s) by default for un-negated variable(s), unless stated otherwise. This 3-SAT variation is known to be NP-complete [Garey and Johnson, 1979]. We denote the instances with satisfiable solutions as

$$\begin{aligned}3\text{-}SAT \;\; = \;\; \{ \; & ((x_i)_{i\in[m]}, (\neg x_i)_{i\in[m]}, \wedge_{i=1}^{c} C_i, C_i = (\vee_{j=1}^{3} x_{i_j}) \\ & \text{or } C_i = (\vee_{j=1}^{3} \neg x_{i_j})) : \text{there exists a satisfiable assignment } \},\end{aligned}$$

where there are m variables (along with its negated variables which are listed explicitly), c clauses, and each clause has three variables or negated variables. A satisfiable assignment is defined to be an assignment of all variables i to zero or one, $x_i \in \{0, 1\}$, such that the boolean formula $\wedge_{i=1}^{c} C_i$ is true or satisfied (i.e., each clause C_i is true or satisfied).

Below, given an instance of 3-SAT

$$\gamma = \big((x_i)_{i\in[m]}, (\neg x_i)_{i\in[m]}, \wedge_{i=1}^{c} C_i, C_i = (\vee_{j=1}^{3} x_{i_j}) \text{ or } C_i = (\vee_{j=1}^{3} \neg x_{i_j})\big)\,,$$

we are going to construct an instance of α-IDS games

$$\beta = \left(([n], (C_i)_{i\in[n]}, (\alpha_i)_{i\in[n]}, (L_i)_{i\in[n]}, (p_i)_{i\in[n]}, (q_{ji})_{j,i\in[n],i\neq j})\right),$$

that correspond to γ.

- There are $n = c+2m$ players: a player for each clause, a player for each variable, and a player for each negated variable. The clause players, variable players, and negated variable players are indexed from 1 to c, $c+1$ to $m+c$, and $m+c+1$ to $2m+c$, respectively.

- First, we find $q \in [0,1]$ such that $1 - \left(1 - \frac{R-p}{1-p-\alpha}\right)^{\frac{1}{2}} > q > 1 - \left(1 - \frac{R-p}{1-p-\alpha}\right)^{\frac{1}{3}}$ for some $0 < 1-\alpha < R < p < 1$, $1 > L > C > 0$, and $R = \frac{C}{L}$. The constraint of the parameters is due to the fact that we want to enforce the condition $1 - p < \alpha$, and to ensure that $0 < \frac{R-p}{1-p-\alpha} < 1$, we require $R - p < 0$ and $R - p > 1 - p - \alpha$. It is not hard to see that such q always exists.

  For each clause player $i \in [c]$ such that $C_i = (\vee_{j=1}^{3} x_{i_j})$, $q_{(i_j+c)i} = q_{i(i_j+c)} = q$ for all j, and for each clause player $i \in [c]$ such that $C_i = (\vee_{j=1}^{3} \neg x_{i_j})$, $q_{(i_j+c+m)i} = q_{i(i_j+c+m)} = q$ for all j. To set the remaining parameters, for each clause player $i \in [c]$, set $C_i = C$, $L_i = L$, $\alpha_i = \alpha$, and $p_i = p$.

  Given the parameters, notice that (1) $\alpha_i > 1 - p_i$ for all i and (2) $(1-q)^2 > \Delta_i^{sc} > (1-q)^3$. Thus, all of the clause players are SC, and each clause player can transfer the "bad event" to its variable players (or negated variable players) and vice versa.

- Using the same q as above, we find $1 > 1 - \alpha' > R' > p' > 0$, $1 > L' > C' > 0$, and $R' = \frac{C'}{L'}$ such that $q > \frac{R'-p'}{1-p'-\alpha'} > 0$. Notice that we can make p' arbitrary small so that $\frac{R'-p'}{1-p'-\alpha'} \approx \frac{R'}{1-\alpha'}$. Moreover, $\frac{R'}{1-\alpha'}$ can be made arbitrary close to zero.

  The constraint of the parameters is due to the fact that we want to enforce the condition $1 - p' > \alpha'$, and to ensure that $0 < \frac{R'-p'}{1-p'-\alpha'} < 1$, we require $R' - p' > 0$ and $R' - p' < 1 - p' - \alpha'$.

  As defined earlier, each variable player $i \in \{c+1, ..., m+c, ..., 2m+c\}$ has transfer risks to and from its clause players (i.e., the clauses in which variable i appears in). In addition of having transfer risks to and from the clauses, each variable player $i \in \{(c+1), ..., (m+c)\}$ has transfer risks from and to its negated variable player, that is $q_{i(i+m)} = q_{(i+m)i} = q$. To set the remaining parameters, for each player $i \in \{c+1, ..., m+c, ..., 2m+c\}$, set $C_i = C'$, $L_i = L'$, $\alpha_i = \alpha'$, and $p_i = p'$.

Given the parameters, notice that (1) $\alpha_i < 1 - p_i$ for all i and (2) $1 > \Delta_i^{ss} > (1-q)$. Thus, all of the variable and negated variable players are SS, and each variable player can transfer the "bad event" to its clause players and its negated variable player (and vice versa).

Moreover, unless defined above, the transfer risks from other clauses to other variables are all zero.

Figure 3.2 depicts the basic structure of the game of a variable. It is easy to see that the construction takes polynomial time.

Lemma 1. *$\gamma \in$ 3-SAT $\implies \beta \in \alpha$-IDS.*

Proof. Given a satisfiable assignment for γ, we show how to construct a PSNE for β. Let $x^{(1)} = \{i \in [m] \ : \ x_i = 1\}$ and $x^{(0)} = \{i \in [m] \ : \ x_i = 0\}$ be the indices of the variables that are assigned a value of one and zero, respectively, in the satisfiable assignment. For consistence, we let a_i to denote the action of any player $i \in [n]$ and construct a PSNE as follows. For each clause player $i \in [c]$, let $a_i = 1$. For each variable player $i \in \{(c+1), ..., (m+c)\}$, if $(i-c) \in x^{(1)}$, $a_i = 1$, otherwise $a_i = 0$. For each negated variable player $i \in \{(m+c+1), ..., (2m+c)\}$ if $(i-(m+c)) \in x^{(0)}$, $a_i = 1$, otherwise $a_i = 0$. Note that by construction $a_i \neq a_{i+m}$ for $i \in \{(c+1), ..., (m+c)\}$. We will call this constructed pure-strategy profile $a = (a_1, ..., a_n)$.

To show that a is a PSNE, we argue that each player is playing its best-response. First, we consider the clause players. Recall that since the clause players are the type of SC, for each $i \in [c]$, we have

$$\mathcal{BR}_i^{sc}(a_{\mathrm{Pa}(i)}) \equiv \begin{cases} \{0\}, & \Delta_i^{sc} > s_i(a_{\mathrm{Pa}(i)}), \\ \{1\}, & \Delta_i^{sc} < s_i(a_{\mathrm{Pa}(i)}), \\ \{0,1\}, & \Delta_i^{sc} = s_i(a_{\mathrm{Pa}(i)}) \ , \end{cases}$$

where $Pa(i) = \{i_1, i_2, i_3\}$ (which corresponds to variables $x_{i_1}, x_{i_2}, x_{i_3}$ of clause i) and $s_i(a_{\mathrm{Pa}(i)}) = \prod_{j \in \mathrm{Pa}(i)} (1-q)^{1-a_j}$. Moreover, by the satisfiable assignment, at least one variable in $\mathrm{Pa}(i)$ is assigned to a value of one which corresponds to at least one variable player that plays action one. Therefore, $(1-q)^2 < s_i(a_{\mathrm{Pa}(i)}) < 1$. By our construction, $(1-q)^2 > \Delta_i^{sc} > (1-q)^3$. It follows that $s_i(a_{\mathrm{Pa}(i)}) > \Delta_i^{sc}$, and the i's best-response is one. This holds for all clause players $i \in [c]$.

For each variable player $i \in \{(c+1), ..., (m+c)\}$, i is a SS player and i's best-response correspondence is

$$\mathcal{BR}_i^{ss}(a_{\mathrm{Pa}(i)}) \equiv \begin{cases} \{0\}, & \Delta_i^{ss} < s_i(a_{\mathrm{Pa}(i)}), \\ \{1\}, & \Delta_i^{ss} > s_i(a_{\mathrm{Pa}(i)}), \\ \{0,1\}, & \Delta_i^{ss} = s_i(a_{\mathrm{Pa}(i)}) \ , \end{cases}$$

where $P(i) = \{m+i\} \cup \{j \in [c] : q_{ji} = q\}$ (its negated variable and the clauses that have variable x_i) and $s_i(a_{\mathrm{Pa}(i)}) = \prod_{j \in \mathrm{Pa}(i)} (1-q)^{1-a_j}$. Since every $i \in [c]$ plays the action one, $s_i(a_{\mathrm{Pa}(i)}) = (1-q)^{1-a_{m+i}}$. Moreover, by our construction, we have $1 > \Delta_i^{ss} > (1-q)$. Therefore, if $a_{m+i} = 1$ then $a_i = 0$, and if $a_{m+i} = 0$ then $a_i = 1$. This is exactly our construction as by the satisfiable assignment either one of them is true but never both. This holds for the negated variable player $i \in \{(m+c+1), ..., (2m+c)\}$. Hence, the pure-strategy profile a is a PSNE. □

Lemma 2. *$\beta \in \alpha$-IDS $\implies \gamma \in$ 3-SAT.*

Proof. Now we show how to construct a satisfiable assignment for γ given a PSNE of β. Let $a = (a_1, ..., a_n)$ be a PSNE of β. First we provide the following claims.

Claim 1. *For every PSNE of β, $a_i = 1$ for all $i \in [c]$.*

Proof. For the sake of contradiction, suppose there is a PSNE such that there is a clause player $i \in [c]$ plays the action zero (i.e., $a_i = 0$). Since $(1-q)^2 > \Delta_i^{sc} > (1-q)^3$, for $a_i = 0$, $\Delta_i^{sc} > (1-q)^3$. It follows that for $j \in \mathrm{Pa}(i) = \{i_1, i_2, i_3\}$, $a_j = 0$. On the other hand, for $j \in \mathrm{Pa}(i)$, $1 > \Delta_j^{ss} > (1-q)$. Since $i \in \mathrm{Pa}(j)$, $s_j(a_{\mathrm{Pa}(j)}) \leq (1-q)$, $a_j = 1$. This is a contradiction, thus our claim holds. □

Claim 2. *For every PSNE of β, $a_i \neq a_{i+m}$ for all $i \in \{c+1, ..., c+m\}$.*

Proof. For the sake of contradiction, we consider the case where there is a PSNE in which either (a) $a_i = a_{i+m} = 0$ or (b) $a_i = a_{i+m} = 1$ for some $i \in \{c+1, ..., c+m\}$. First, we consider the case of (a) where $a_i = a_{i+m} = 0$

for some $i \in \{c+1, ..., c+m\}$. By construction, we have $1 > \Delta_i^{ss} > (1-q)$ and $1 > \Delta_{i+m}^{ss} > (1-q)$. It follows that either i or $m+i$ is not playing the best-response; that is, $s_i(a_{\text{Pa}(i)}) \leq (1-q)$ or $s_{m+i}(a_{\text{Pa}(m+i)}) \leq (1-q)$ since $(m+i) \in \text{Pa}(i)$ and $i \in \text{Pa}(m+i)$.

Now suppose that we consider (b) where $a_i = a_{i+m} = 1$ for some $i \in \{c+1, ..., c+m\}$. From Claim 1, all $i \in [c]$, $a_i = 1$. It follows that $s_i(a_{\text{Pa}(i)}) = (1-q)^{1-a_{m+i}}$ and $s_{m+i}(a_{\text{Pa}(m+i)}) = (1-q)^{1-a_i}$. Hence, i or $m+i$ is not playing the best-response. Thus, our claim holds. □

Claim 3. *For every PSNE of β, for each $i \in [c]$, there is a $j \in Pa(i)$ such that $a_j = 1$.*

Proof. For the sake of contradiction, suppose that there is a PSNE where there is $i \in [c]$ such that $a_i = 1$, and for all $j \in \text{Pa}(i)$ $a_j = 0$. Notice that by construction, $(1-q)^2 > \Delta_i^{sc} > (1-q)^3$ and $a_i = 1$ whenever $s_i(a_{\text{Pa}(i)}) > (1-q)^2$. However, $s_i(a_{\text{Pa}}i) = (1-q)^3$ since $a_j = 0$ for all $j \in \text{Pa}(i)$. This is a contradiction and our claim holds. □

Given the claims, we construct a satisfiable assignment for γ as follows. For each $i \in \{(c+1), ..., (c+m)\}$ if $a_i = 1$, set $x_{(i-c)} = 1$, otherwise, set $x_{(i-c)} = 0$. For each $i \in \{(c+m+1), ..., (2m+c)\}$, $a_i = 1$, set $\neg x_{(i-(c+m))} = 1$, otherwise, set $\neg x_{(i-(c+m))} = 0$. From Claim 2, we know that $x_i \neq \neg x_i$ for all $i \in [m]$. For each clause C_i for $i \in [c]$, by Claim 3, there is at least one variable that makes C_i true. Moreover, since all of the clauses are true (Claim 1), we have that the PSNE yields a satisfiable assignment. □

It is easy to see that given a pure-strategy profile, we can verify whether it is a PSNE of an α-IDS game in polynomial time. This fact, together with Lemma 1 and Lemma 2, we have our hardness result. □

3.3.1 SC α-IDS games

What is the complexity of determining whether a PSNE exists in SC α-IDS games (i.e. $\alpha_i > 1 - p_i$)? It turns out that SC players have the characteristics of following the actions of other players. If there are enough SC players who take the action invest, then some remaining SC player(s) will follow suit. This is evident from the safety function and the best-response condition. Consider the dynamics in which everybody starts off with playing the action of not invest. If there are some players that are not best-responding, then their best strategy is play the action invest. We can safely change the actions of those players to invest. Then, for the remaining players, we continue to check to see if any of them is not best-responding. If not, we have a PSNE, otherwise, we change the action of the not best-responding players to invest. The process continues until we have reached a PSNE.

Theorem 6. *There is an $O(n^2)$-time algorithm to compute a PSNE of any n-player SC α-IDS game.*

Note that once a player plays invest, other players will either stay and play the action of not invest or change his/her action to invest. The players that play the action not invest do not affect the strategy of the players that already have decided to invest. Players that have decided to invest will continue to invest because only more players will invest.

3.3.2 SS α-IDS games

Unlike the SC case, an SS α-IDS game may not have a PSNE when $n > 2$.

Proposition 2. *Suppose we have an n-player SS α-IDS game with $1 > \Delta_i^{ss} > (1 - q_{ji})$ where j is the parent of i. (a) If the game graph is a directed tree, then the game has a PSNE. (b) If the game graph is a a directed cycle, then the game has a PSNE if and only if n is even.*

Proof. (a) The root of the tree will always play no-invest while the immediate children of the root will always play invest at a PSNE. Moreover, assigning the action invest or no-invest to any node that has an odd or even (undirected) distance to the root, respectively, completes the PSNE.

(b) For even n, an assignment in which any independent set of $\frac{n}{2}$ players play invest form a PSNE. For odd n, suppose there is a PSNE in which I players invest and N players do not invest, such that $I + N = n$. The investing players must have I parents that do not invest and the non-investing players must have N parents that play invest. Moreover, $I \leq N$ and $N \leq I$ implies that $I = N$. Hence, an odd n cycle cannot have a PSNE. □

3.4 Computing all NE in α-IDS games

Given that we can compute a PSNE in SC α-IDS games in polynomial time, we now study whether we can compute *all* MSNE of α-IDS games. We prove that we can compute all MSNE in polynomial time in the case of uniform-transfer SC α-IDS games, and a subset of all MSNE in the case of uniform SS and SC+SS α-IDS games. A *uniform transfer α-IDS game* is an α-IDS game where the transfer probability to another players from a particular player is the same regardless of the destination. In other words, the players do not discriminate the destinations of the transfers. More formally, $q_{ij} = \delta_i > 0$ for all players i and j $(i \neq j)$. Hence, we have a complete graph with bidirectional transfer probabilities. We can express the overall safety function given a mixed-strategy profile $x \in [0,1]^n$ as $s(x) = \prod_{i=1}^{n} [1 - (1 - x_i)\delta_i]$.

Recall that whether a player, say player w, plays a particular strategy depends on whether Δ_w^{SC} (or Δ_w^{SS}) is is less than, greater than, or equal to $s_w(x_{\text{Pa}(i)})$. Given the transfer probabilities, we have that $\text{Pa}(w) = [n]$. Without loss of generality, we can multiple Δ_w^{SC} (or Δ_w^{SS}) and $s_w(x_{\text{Pa}(i)})$ by $(1 - (1 - x_w)\delta_w)$ so that we only need to compare the values of $\Delta_w^{SC}(1 - (1 - x_w)\delta_w)$ (or $\Delta_w^{SC}(1 - (1 - x_w)\delta_w)$) and $s(x)$. Now, we can determine the best-response of SC or SS player exactly based solely on the values of $\Delta_i^{sc}(1 - (1 - x_i)\delta_i)$, for SC, relative to $s(x)$; similarly for SS.

We assume, without loss of generality, that for all players i, $R_i > 0$, $\delta_i > 0$, $p_i > 0$, and $\alpha_i > 0$. Given a mixed-strategy profile x, we partition the players by type with respect to x: let $I \equiv I(x) \equiv \{i \mid x_i = 1\}$, $N \equiv N(x) \equiv \{i \mid x_i = 0\}$, and $P \equiv P(x) \equiv \{i \mid 0 < x_i < 1\}$ be the set of players that, with respect to x, *fully invest* in protection, *do not invest* in protection, and *partially invest* in protection, respectively.

3.4.1 Uniform-transfer SC α-IDS games

The results of this section are non-trivial extensions of those of Kearns and Ortiz [2004]. In particular, we can construct a polynomial-time algorithm to compute all MSNE of a uniform-transfer SC α-IDS game, along the same lines of Kearns and Ortiz [2004], by extending their Ordering Lemma (their Lemma 3) and Partial-Ordering Lemma (their Lemma 4). [3] We now present our versions of the lemmas.

Lemma 3 (Ordering Lemma). *Suppose x is a NE of a uniform-transfer SC α-IDS game. Then for any $i \in I$ (investing players), any $j \in P$ (partially*

[3] Take their R_i/p_i's and replace them with our corresponding Δ_i^{sc}'s.

investing players), and any $k \in N$ (not investing players), then

$$\begin{aligned} \Delta_i^{sc} &\leq \Delta_j^{sc} \\ \Delta_i^{sc} &\leq (1-\delta_k)\Delta_k^{sc} < \Delta_k^{sc} \\ (1-\delta_j)\Delta_j^{sc} &\leq (1-\delta_k)\Delta_k^{sc} \end{aligned}$$

Proof. The inequalities follow immediately by using the overall safety function to compare the players in I, P, and N. In particular, let $i \in I$ be an investing player, $j \in P$ be a partially investing player, and $k \in N$ be a not investing player. It follows that

$$\Delta_i^{SC} = \Delta_i^{SC}(1-(1-x_i)\delta_i) \leq s(x) = \Delta_j^{SC}(1-(1-x_j)\delta_j) \leq \Delta_j^{SC},$$

where the first equality is $x_i = 1$, the first inequality and the second equality are by the best-response correspondence of a SC player, and the last inequality is because $(1-(1-x_j)\delta_j)$ is between zero and one. This condition gives us the first inequality.

The second inequality follows that

$$\Delta_i^{SC} = \Delta_i^{SC}(1-(1-x_i)\delta_i) \leq s(x) \leq \Delta_k^{SC}(1-(1-x_k)\delta_k) < \Delta_k^{SC},$$

where this is according to the best-response correspondence, $x_k = 0$, and $(1-\delta_k)$ is strictly between zero and one.

Finally, the last inequality follows that

$$\Delta_j^{SC}(1-(1-x_j)\delta_j) = s(x) \leq \Delta_k^{SC}(1-(1-x_k)\delta_k) = \Delta_k^{SC}(1-\delta_k),$$

which, again, is by the best-response correspondence condition. □

The following Lemma specifies the strategies of the players in the partially investing set.

Lemma 4 (Partial Investment Lemma)**.** *Suppose x is a NE of a uniform-transfer SC α-IDS game. For any $j \in P$,*

1. *If $|P| = 1$, then $x_j \in \frac{1}{\delta}(\frac{1}{\Delta_j^{sc}}V - (1-\delta_j))$*
2. *if $|P| > 1$, then $x_j = \frac{1}{\delta}(\frac{1}{\Delta_j^{sc}}V^* - (1-\delta_j))$*

where $V = [\max_{i \in I} \Delta_i^{sc}, \min_{k \in N}(1-\delta_k)\Delta_k^{sc}]$ and $V^* = \left(\frac{\prod_{j \in P} \Delta_j^{sc}}{\prod_{k \in N}(1-\delta_k)}\right)^{\frac{1}{|P|-1}}$.

Proof. Suppose that $|P| = 1$. By the best-response condition $\Delta_j^{sc} = \prod_{l\in N}(1-\delta_l)$. Moreover

$$\forall\ i \in I,\, \Delta_i^{sc} \leq (1-(1-x_j)\delta_j)\prod_{l\in N}(1-\delta_l)$$

and

$$\forall\ k \in N,\, (1-\delta_k)\Delta_k^{sc} \geq (1-(1-x_j)\delta_j)\prod_{l\in N}(1-\delta_l).$$

If we solve for x_j, we can obtain the values that x_j can take at an equilibrium.

Suppose that $|P| > 1$. By the best-response condition

$$\Delta_j^{sc} = \prod_{p\in P-\{j\}}(1-(1-x_p)\delta_p)\prod_{l\in N}(1-\delta_l)\ \forall j \in P.$$

Furthermore, for $j \in P$,

$$\prod_{k\in P-j}\Delta_k^{sc} = \\ (1-(1-x_j)\delta_j)^{|P|-1}\left(\prod_{p\in P-j}(1-(1-x_p)\delta_p)\right)^{|P|-2}\left(\prod_{l\in N}(1-\delta_l)\right)^{|P|-1}$$

It follows that

$$\frac{\prod_{k\in P-j}\Delta_k^{sc}}{\left(\prod_{p\in P-j}(1-(1-x_p)\delta_p)\right)^{|P|-2}\left(\prod_{l\in N}(1-\delta_l)\right)^{|P|-1}} = (1-(1-x_j)\delta_j)^{|P|-1}$$

$$\frac{\prod_{k\in P}\Delta_k^{sc}}{\left(\prod_{p\in P-j}(1-(1-x_p)\delta_p)\right)^{|P|-1}\left(\prod_{l\in N}(1-\delta_l)\right)^{|P|}} = (1-(1-x_j)\delta_j)^{|P|-1}$$

$$\left(\frac{\prod_{k\in P}\Delta_k^{sc}}{\left(\prod_{p\in P-j}(1-(1-x_p)\delta_p)\right)^{|P|-1}\left(\prod_{l\in N}(1-\delta_l)\right)^{|P|}}\right)^{\frac{1}{|P|-1}} = (1-(1-x_j)\delta_j)$$

$$\frac{\left(\frac{\prod_{k\in P}\Delta_k^{sc}}{\prod_{l\in N}(1-\delta_l)}\right)^{\frac{1}{|P|-1}}}{\left(\prod_{p\in P-j}(1-(1-x_p)\delta_p)\right)\prod_{l\in N}(1-\delta_l)} = (1-(1-x_j)\delta_j)$$

$$\left(\frac{\prod_{k\in P}\Delta_k^{sc}}{\prod_{l\in N}(1-\delta_l)}\right)^{\frac{1}{|P|-1}}\frac{1}{\Delta_j^{sc}} = (1-(1-x_j)\delta_j)$$

The result follows from solving for x_j. □

Algorithm 1: Compute all Nash equilibria of SC α-IDS games

```
Input   : An instance of $n$-players SC $\alpha$-IDS Game
Output: S - The set of all Nash equilibria of the input game
$I \leftarrow \{1, ..., n\}, P \leftarrow \{\}, N \leftarrow \{\}$
$S \leftarrow$ TestNash$(I, P, N)$
Order $(i_1, i_2, ..., i_n)$ such that $\Delta^{sc}_{i_1} \geq ... \geq \Delta^{sc}_{i_n}$
foreach $k = 1, ..., n$ do
    $P \leftarrow P \cup \{i_k\}$, $I \leftarrow I - \{i_k\}$, $N \leftarrow \{\}$, $S \leftarrow S \bigcup$ TestNash$(I, P, N)$
    Let $P' \leftarrow P$ and order $(j_1, ..., j_k)$ such that
    $(1 - \delta_{j_1})\Delta^{sc}_{j_1} \geq ... \geq (1 - \delta_{j_k})\Delta^{sc}_{j_k}$
    foreach $m = 1, ..., k$ do
        $N \leftarrow N \cup \{j_m\}, P' \leftarrow P' - \{j_m\}$ $S \leftarrow S \bigcup$ TestNash$(I, P', N)$
    end foreach
end foreach
return $S$
```

Algorithm 2: TestNash subroutine

```
Input   : A partition of the players into I, P, and N
Output: S - The set of all Nash equilibria consistent with the input
          partition
$\forall i \in I$, $x_i \leftarrow 0$, $\forall k \in N$, $x_k \leftarrow 0$
if $|P| = 1$ and $j \in P$ (Lemma 4 Part 1) then
    Let $U' = U \cap (0, 1)$
    if $\Delta^{sc}_j = \prod_{k \in N}(1 - \delta_k)$ and $U' \neq \emptyset$ then
        $S \leftarrow \{\mathbf{y} \mid y_j \in U', \mathbf{y}_{-j} = \mathbf{x}_{-j}\}$
    end if
else Lemma 4 Part 2
    $\forall j \in P$, compute $x_j$
    if $\mathbf{x}$ is an MSNE of the input game then
        $S \leftarrow \{x\}$
    end if
end if
return $S$
```

Algorithm 1, constructed based on the above characterizations, compute all NE in uniform-transfer SC α-IDS games. The subroutine **TestNash** of Algorithm 1 is outlined in Algorithm 2.

The running time of Algorithm 1 is $O(n_{sc}^3)$ where the **TestNash** subroutine takes $O(n)$, and line 7 of the algorithms runs in $O(n(1+2+...+n) = O(n^3)$ times where $n = n_{sc}$ for SC players. This is similar to running-time analysis of traditional uniform-transfer IDS games done by Kearns and Ortiz [2004].

Theorem 7. *There exists an $O(n^4)$-time algorithm to compute all MSNE of an uniform-transfer n-player SC α-IDS game.*

The significance of the theorem lies in its simplicity. That we can extend almost the same computational results, and structural implications on the solution space, to a considerably more general, and perhaps even more realistic model.

3.4.2 Uniform-transfer SS α-IDS games

Unlike the SC case, the ordering we get for the SS case does not yield an analogous lemma. Nevertheless, it turns out that we can still determine the mixed strategies of the partially-investing players in P relative to a partition. The result is a Partial-Investment Lemma that is analogous to that of Kearns and Ortiz [2004] for traditional IDS games. [4]

Lemma 5. *(Partial Investment Lemma) Suppose x is a NE of a uniform-transfer SS α-IDS game. For any $j \in P$,*

1. *If $|P| = 1$, then $x_j \in \frac{1}{\delta}(\frac{1}{\Delta_j^{ss}}V - (1 - \delta_j))$*
2. *if $|P| > 1$, then use Lemma 4 part 2.*

where $V = [\max_{k \in N}(1 - \delta_k)\Delta_k^{ss}, \min_{i \in I} \Delta_i^{ss}]$.

Proof. The proof is similar to the one in Lemma 4. □

Indeed, a naive way to compute all NE is to consider all of the possible combinations of players into the investment, partial investment, and not investment sets and apply the Partial-Investment Lemma alluded to in the

[4] Take their Lemma 4 and replace R_i/p_i there by Δ_i^{ss} here, and replace the expression for V there by $V \equiv [\max_{k \in N}(1 - \delta_k)\Delta_k^{ss}, \min_{i \in I} \Delta_i^{ss}]$.

previous paragraph to compute the mixed strategies. However, this would take $O(n_{ss}3^{n_{ss}})$ worst-case time to compute any equilibrium. So, how can we efficiently perform this computation? As mentioned earlier, SS players are less likely to invest when there is a large number of players investing and have "opposite" behavior as the SC players (i.e., the best response is flipped). Hence, imposing a "flip" ordering (Ordering 1) that is opposite of the SC case seems natural. If we assume such a specific ordering of the players at equilibrium, then we can compute all NE consistent with that specific ordering efficiently, as we discuss earlier for the SC case. Mirroring the SC α-IDS game, we settle for computing all NE that satisfy the following ordering.

Ordering 1. *For all $i \in I, j \in P$, and $k \in N$,*

$$\begin{aligned}
&(1-\delta_k)\Delta_k^{ss} \le (1-\delta_j)\Delta_j^{ss} < \Delta_j^{ss} \\
&(1-\delta_j)\Delta_j^{ss} \le \Delta_j^{ss} \le \Delta_i^{ss} \\
&(1-\delta_k)\Delta_k^{ss} \le (1-\delta_i)\Delta_i^{ss} \le \Delta_i^{ss}
\end{aligned}$$

The first and last set of inequalities (ignoring the middle one) follow from the consistency constraint imposed by the overall safety function. The middle set of inequalities restrict and reduce the number of possible NE configurations we need to check. It is possible that the $(1-\delta_k)\Delta_k^{ss} > (1-\delta_j)\Delta_j^{ss}$ or $(1-\delta_k)\Delta_k^{ss} > (1-\delta_i)\Delta_i^{ss}$ at an NE, but we do not consider those types of NE. Our hardness results presented in the upcoming Section 3.4.4 suggest that, in general, computing all MSNE without any of the constraints (in general graph structures) above is likely hard.

Theorem 8. *There exists an $O(n^4)$-time algorithm to compute all MSNE consistent with Ordering 1 of an uniform-transfer n-player SS α-IDS game.*

The algorithm is provided below. Note that the subroutine **TestNash** of Algorithm 3 can be constructed similarly from Algorithm 2 where it will use Lemma 5.

3.4.3 Uniform-transfer SC+SS α-IDS games

For the uniform variant of the SC+SS α-IDS games, we could partition the players into either SC or SS and modify the respective algorithms to compute all NE. Unfortunately, this is computationally infeasible because we can only

Algorithm 3: Compute All Nash Equilibria of SS consistent with Ordering 1

```
Input  : An instance of $n$-players SS $\alpha$-IDS Game
Output : S - A set of all Nash Equilibria that is consistent with
         Ordering 1
$I \leftarrow \{\}, P \leftarrow \{\}, N \leftarrow \{1, ..., n\}$
$S \leftarrow$ TestNash($I, P, N, S$)
Order $(i_1, i_2, ..., i_n)$ such that $(1 - \delta_{i_1})\Delta^{ss}_{i_1} \geq ... \geq (1 - \delta_{i_n})\Delta^{ss}_{i_n}$
foreach $k = 1, ..., n$ do
    $P \leftarrow P \cup \{i_k\}$, $N \leftarrow N - \{i_k\}$, $I \leftarrow \{\}$, $S \leftarrow$ TestNash($I, P, N, S$)
    Let $P' \leftarrow P$ and order $(j_1, ..., j_k)$ such that $\Delta^{ss}_{j_1} \geq ... \geq \Delta^{ss}_{j_k}$
    foreach $m = 1, ..., k$ do
        $I \leftarrow I \cup \{j_m\}, P' \leftarrow P' - \{j_m\}$ $S \leftarrow$ TestNash($I, P', N, S$)
    end foreach
end foreach
return $S$
```

compute all NE in polynomial time in the SC case. Again, if we settle for computing all NE consistent with Ordering 1, then we can devise an efficient algorithm. *From now on, the fact that we are only considering NE consistent with Ordering 1 is implicit, unless noted otherwise.* The idea is to partition the players into a class of SC and a class of SS players. From the characterizations stated earlier, it is clear that there are only a polynomial number of possible partitions we need to check for each class of players. Since the ordering results are based on the same overall safety function, the orderings of SC and SS players do not affect each other. Hence, without loss of generality, starting with the algorithm described earlier as a based routine for SC players, we do the following. For each possible equilibrium configuration of the SC players, we first run the algorithm described in the previous section for SS players and then test whether the resulting mixed-strategy profile is a NE. This guarantees that we check every possible equilibrium combination. A running-time analysis yields our next result.

Theorem 9. *There exists an $O(n_{sc}^4 n_{ss}^3 + n_{sc}^3 n_{ss}^4)$-time algorithm to compute all NE consistent with Ordering 1 of an uniform-transfer n-player SC+SS α-IDS game, where $n = n_{sc} + n_{ss}$.*

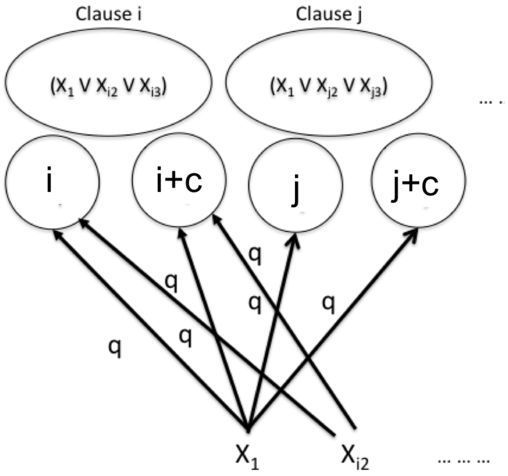

Clause i
Clause j
$(X_1 \vee X_{i2} \vee X_{i3})$
$(X_1 \vee X_{j2} \vee X_{j3})$
.... .
i
i+c
j
j+c
q
q
q
q
q
q
X_1
X_{i2}
... ...

Figure 3.3: **Monotone 1 In 3-SAT-induced SC α-IDS game-graph.**

3.4.4 Computing all MSNE of arbitrary α-IDS games is intractable, in general

In this section, we prove that determining whether there exists a PSNE consistent with a partial-assignment of the actions to some players is NP-complete, even if the transfer probability takes only two values: $\delta_i \in \{0, q\}$ for some $q \in (0, 1)$.

We consider the *Pure-Nash-Extension problem* [Kearns and Ortiz, 2004] for binary-action n-player games that takes as input a description of the game and a *partial* assignment $a \in \{0, 1, *\}^n$. We want to know whether there is a *complete* assignment $b \in \{0, 1\}^n$ consistent with a. Indeed, computing all NE is at least as difficult as the Pure-Nash Extension problem.

Theorem 10. *The Pure-Nash-Extension problem for n-player SC α-IDS games is NP-complete.*

Proof. We first define the notations that will be used in the proof. In particular, we consider the problem of determining whether there is a PSNE in SC α-IDS games while fixing some actions of some players. More specifically, we denote the instances with PSNE as

$$\begin{aligned}\text{SC } \alpha\text{-}IDS \;=\; & \{\; ([n], (C_i)_{i\in[n]}, (\alpha_i)_{i\in[n]}, (L_i)_{i\in[n]}, (p_i)_{i\in[n]}, (q_{ji})_{j,i\in[n],i\neq j},\\ & (a_i)_{i\in S} \subseteq \{0,1\}^{|S|}) : \text{there exists a PSNE in } \mathbb{G} \text{ with}\\ & \qquad \text{the players in } S \text{ play according to } (a_i)_{i\in S} \;\}.\end{aligned}$$

We will reduce our problem from Monotone 1 in 3-SAT where each clause of the 3-SAT has exactly three variables and consists of (un-negated) variables. We use the term variable(s) by default for un-negated variable(s), unless stated otherwise. The solution to the Monotone 1 in 3-SAT is to find a satisfiable assignment such that exactly one variable is true in each clause. The Monotone 1 in 3-SAT is known to be NP-complete [Garey and Johnson, 1979]. We denote the instances with satisfiable solutions as

$$\begin{aligned}\text{M 1 in 3-}SAT \;=\; & \{\; ((x_i)_{i\in[m]}, \wedge_{i=1}^{c} C_i, C_i = (\vee_{j=1}^{3} x_{i_j})) : \text{there exists a}\\ & \text{satisfiable assignment with exactly one}\\ & \text{variable true in each clause } \},\end{aligned}$$

where there are m variables, c clauses, and each clause has three (un-negated) variables. A satisfiable assignment is defined to be an assignment of all variables i to zero or one, $x_i \in \{0,1\}$, such that the boolean formula $\wedge_{i=1}^{c} C_i$ is true or satisfied (i.e., each clause C_i is true or satisfied and has exactly one variable true).

Below, given an instance of Monotone 1 in 3-SAT

$$\gamma = \left((x_i)_{i\in[m]}, \wedge_{i=1}^{c} C_i, C_i = (\vee_{j=1}^{3} x_{i_j})\right),$$

we are going to construct an instance of SC α-IDS games with partial assignments

$$\beta = \left([n], (C_i)_{i\in[n]}, (\alpha_i)_{i\in[n]}, (L_i)_{i\in[n]}, (p_i)_{i\in[n]}, (q_{ji})_{j,i\in[n],i\neq j}, (a_i)_{i\in S} \subseteq \{0,1\}^{|S|}\right),$$

that correspond to γ.

- There are $n = 2c + m$ players: two players for each clause and a player for each variable. The clause players and the variable players are indexed from 1 to $2c$ and $2c+1$ to $2c+m$, respectively.

- First, we find $q \in [0,1]$ such that $1 - \left(1 - \frac{R-p}{1-p-\alpha}\right)^{\frac{1}{2}} > q > 1 - \left(1 - \frac{R-p}{1-p-\alpha}\right)^{\frac{1}{3}}$ for some $0 < 1 - \alpha < R < p < 1$, $1 > L > C > 0$, and $R = \frac{C}{L}$. The constraint of the parameters is due to the fact that we want to enforce the condition $1 - p < \alpha$, and to ensure that $0 < \frac{R-p}{1-p-\alpha} < 1$, we require $R - p < 0$ and $R - p > 1 - p - \alpha$. It is not hard to see that such q always exists.

  For each clause player $i \in [c]$ such that $C_i = (\vee_{j=1}^{3} x_{i_j})$, $q_{(i_j+2c)i} = q$ for all j. To set the remaining parameters, for each clause player $i \in [c]$, set $C_i = C$, $L_i = L$, $\alpha_i = \alpha$, and $p_i = p$.

  Given the parameters, notice that (1) $\alpha_i > 1 - p_i$ for all i and (2) $(1-q)^2 > \Delta_i^{sc} > (1-q)^3$. Thus, all of the clause players in $[c]$ are SC, and each clause player has transfer risks from its variable players.

- Using the same q as above, we find $0 < 1 - \alpha' < R' < p' < 1$, $1 > L' > C' > 0$, and, $R' = \frac{C'}{L'}$ such that $(1-q) > 1 - \frac{R'-p'}{1-p'-\alpha'} > (1-q)^2$. The constraint of the parameters are based on the same reasoning as above. Notice that for each possible of $q \in (0,1)$, we can find $\frac{R'-p'}{1-p'-\alpha'}$ such that $1 - \left(1 - \frac{R'-p'}{1-p'-\alpha'}\right)^{\frac{1}{2}} < q < \frac{R'-p'}{1-p'-\alpha'}$ (i.e., fixing the value of p' and find $(1-\alpha')$ and R' arbitrary close or far from p').

  For each clause player $i \in \{c+1, ..., 2c\}$ such that $C_{i-c} = \left(\vee_{j=1}^{3} x_{(i-c)_j}\right)$, $q_{((c-i)_j+2c)i} = q$ for all j. To set the remaining parameters, for each clause player $i \in \{c+1, ..., 2c\}$, set $C_i = C'$, $L_i = L'$, $\alpha_i = \alpha'$, and $p_i = p'$.

  Note that the clause players here are also SC players.

- Find $1 > p'' > 0$, $1 > R'' > 0$, $1 > L'' > C'' > 0$, and $R'' = \frac{C''}{L''}$ such that $R'' = p''$. For each variable player $i \in \{2c+1, ..., 2c+m\}$, $C_i = C'''$, $L_i = L''$, $p_i = p''$, and $\alpha_i = \alpha''$.

  The variable players are indifferent from playing the action invest or not invest and are SC players.

- Here, we construct a partial action profile for some of the players. In particular, for each clause player $i \in [c]$, $a_i = 1$ and $a_{i+c} = 0$. Thus, we are giving a partial action profile of all clause players.

Moreover, unless defined above, the transfer risks from other clauses to other variables are all zero.

Figure 3.3 depicts the basic structure of the clauses and variables. It is easy to see that the construction takes polynomial time.

Lemma 6. *$\gamma \in$ M 1 in 3-SAT $\implies \beta \in$ SC α-IDS.*

Proof. Given a satisfiable assignment for γ, we show how to construct a PSNE for β. Let $x^{(1)} = \{i \in [m] \ : \ x_i = 1\}$ be the indices of the variables that are assigned a value of one in the satisfiable assignment. For consistence, we let a_i to denote the action of any player $i \in [n]$ and construct a PSNE as follows. For each of the variable player $i \in \{2c+1, ..., 2c+m\}$, $a_i = 1$ if $(i-2c) \in x^{(1)}$ and $a_i = 0$ otherwise. Together with the partial action profile of the clauses, we will call this constructed pure-strategy profile $a = (a_1, ..., a_n)$.

To show that a is a PSNE, we argue that each player is playing its best-response. First, we consider the clause players. Recall that since the clause players are the type of SC, for each $i \in [c]$, we have

$$\mathcal{BR}_i^{sc}(a_{\text{Pa}(i)}) \equiv \begin{cases} \{0\}, & \Delta_i^{sc} > s_i(a_{\text{Pa}(i)}), \\ \{1\}, & \Delta_i^{sc} < s_i(a_{\text{Pa}(i)}), \\ \{0,1\}, & \Delta_i^{sc} = s_i(a_{\text{Pa}(i)}) \ , \end{cases}$$

where $Pa(i) = \{i_1, i_2, i_3\}$ (which corresponds to variables $x_{i_1}, x_{i_2}, x_{i_3}$ of clause i) and $s_i(a_{\text{Pa}(i)}) = \prod_{j \in \text{Pa}(i)} (1-q)^{1-a_j}$. Moreover, by the satisfiable assignment, exactly one variable in Pa(i) is assigned to a value of one which corresponds to exactly one variable player that plays action one. Therefore, $s_i(a_{\text{Pa}(i)}) = (1-q)^2$. By our construction, $(1-q)^2 > \Delta_i^{sc} > (1-q)^3$. It follows that $s_i(a_{\text{Pa}(i)}) > \Delta_i^{sc}$, and the i's best-response is one. This holds for all clause players $i \in [c]$. On the other hand, for the clause player $i \in \{c+1, ..., 2c\}$, $s_i(a_{\text{Pa}(i)}) = (1-q)^2$ as well. Since clause player i is also a SC player, we have the same best-response correspondence. By our construction, $(1-q) > \Delta_i^{sc} > (1-q)^2$, it follows that $\Delta_i^{sc} > s_i(a_{\text{Pa}(i)})$ and $a_i = 0$ is the best-response.

For each variable player $i \in \{2c+1, ..., 2c+m\}$, i has no parent and i's overall risk is 0. To determine whether i plays the action invest or not invest, we only need to compare the value of R_i and p_i. By construction, $R_i = p_i$ for all variable players i, we have that the variable players are indifferent between playing one and zero. Hence, the pure-strategy profile a is a PSNE. □

Lemma 7. *$\beta \in$ SC α-IDS $\implies \gamma \in$ M 1 in* 3-*SAT*.

Proof. Now we show how to construct a satisfiable assignment for γ given a PSNE of β. Let $a = (a_1, ..., a_n)$ be a PSNE of β. For each variable $i \in [m]$, if $a_{2m+i} = 1$ then $x_i = 1$ and if $a_{2m+i} = 0$ then $x_i = 0$. To show that each clause, say $i \in [c]$, has exactly one variable that is true, we observe the best-response of clause players i and $c + i$ that correspond to clause i. Given the fixed action of $a_i = 1$ and $a_{c+i} = 0$ at a PSNE, it implies that $s_i(a_{\mathrm{Pa}(i)}) > \Delta_i^{sc}$ and $s_{c+i}(a_{\mathrm{Pa}(c+i)}) < \Delta_{c+i}^{sc}$. Since $(1 - q)^2 > \Delta_i^{sc} > (1 - q)^3$, $(1 - q) > \Delta_{c+i}^{sc} > (1 - q)^2$, $\mathrm{Pa}(c + i) = \mathrm{Pa}(i)$, $|Pa(i)| = 3$, and the transfer risks are the same, we have $s_{c+i}(a_{\mathrm{Pa}(c+i)}) = (1-q)^2$. This implies that exactly one of the variables is true. □

It is easy to see that given a (partial) pure-strategy profile, we can verify whether it is a PSNE of a SC α-IDS game in polynomial time. This fact, together with Lemma 6 and Lemma 7, we have our hardness result.

□

A similar proof argument yields the following computational-complexity result.

Theorem 11. *The Pure-Nash Extension problem for n-player SS α-IDS games is NP-complete.*

Proof. This is similar to the proof of Theorem 10 except the best-response of the players and using the game graph as in Figure 3.3. For ease of notations, we will use the same notations defined in the proof of Theorem 10.

Below, given an instance of Monotone 1 in 3-SAT

$$\gamma = \left((x_i)_{i\in[m]}, \wedge_{i=1}^{c} C_i, C_i = (\vee_{j=1}^{3} x_{i_j})\right),$$

we are going to construct an instance of SS α-IDS games with partial assignments

$$\beta = \left([n], (C_i)_{i\in[n]}, (\alpha_i)_{i\in[n]}, (L_i)_{i\in[n]}, (p_i)_{i\in[n]}, (q_{ji})_{j,i\in[n],i\neq j}, (a_i)_{i\in S} \subseteq \{0,1\}^{|S|}\right),$$

that correspond to γ.

- There are $n = 2c + m$ players: two players for each clause and a player for each variable. The clause players and the variable players are indexed from 1 to $2c$ and $2c + 1$ to $2c + m$, respectively.

- First, we find $q \in [0,1]$ such that $1 - \left(1 - \frac{R-p}{1-p-\alpha}\right)^{\frac{1}{2}} > q > 1 - \left(1 - \frac{R-p}{1-p-\alpha}\right)^{\frac{1}{3}}$ for some $1 > 1-\alpha > R > p > 0$, $1 > L > C > 0$, and $R = \frac{C}{L}$. The constraint of the parameters is due to the fact that we want to enforce the condition $1 - p > \alpha$, and to ensure that $0 < \frac{R-p}{1-p-\alpha} < 1$, we require $R - p > 0$ and $R - p < 1 - p - \alpha$. It is not hard to see that such q always exists.

  For each clause player $i \in [c]$ such that $C_i = (\vee_{j=1}^{3} x_{i_j})$, $q_{(i_j+2c)i} = q$ for all j. To set the remaining parameters, for each clause player $i \in [c]$, set $C_i = C$, $L_i = L$, $\alpha_i = \alpha$, and $p_i = p$.

  Given the parameters, notice that (1) $\alpha_i < 1 - p_i$ for all i and (2) $(1-q)^2 > \Delta_i^{ss} > (1-q)^3$. Thus, all of the clause players in $[c]$ are SS, and each clause player has transfer risks from its variable players.

- Using the same q as above, we find $1 > 1 - \alpha' > R' > p' > 0$, $1 > L' > C' > 0$, and $R' = \frac{C'}{L'}$ such that $(1-q) > 1 - \frac{R'-p'}{1-p'-\alpha'} > (1-q)^2$. The constraint of the parameters are based on the same reasoning as above. Notice that for each possible of $q \in (0,1)$, we can find $\frac{R'-p'}{1-p'-\alpha'}$ such that $1 - \left(1 - \frac{R'-p'}{1-p'-\alpha'}\right)^{\frac{1}{2}} < q < \frac{R'-p'}{1-p'-\alpha'}$ (i.e., fixing the value of p' and find $(1-\alpha')$ and R' arbitrary close or far from p').

  For each clause player $i \in \{c+1, ..., 2c\}$ such that $C_{i-c} = \left(\vee_{j=1}^{3} x_{(i-c)_j}\right)$, $q_{((c-i)_j+2c)i} = q$ for all j. To set the remaining parameters, for each clause player $i \in \{c+1, ..., 2c\}$, set $C_i = C'$, $L_i = L'$, $\alpha_i = \alpha'$, and $p_i = p'$.

  Note that the clause players here are also SS players.

- Find $1 > 1 - \alpha'' > R'' > p'' > 0$, $1 > L'' > C'' > 0$, and $R'' = \frac{C''}{L''}$ such that $R'' = p''$. For each variable player $i \in \{2c+1, ..., 2c+m\}$, $C_i = C''$, $L_i = L''$, $p_i = p''$, and $\alpha_i = \alpha''$.

  The variable players are indifferent from playing the action invest or not invest and are SS players.

- Here, we construct a partial action profile for some of the players. In particular, for each clause player $i \in [c]$, $a_i = 0$ and $a_{i+c} = 1$. Thus, we are giving a partial action profile of all clause players.

Moreover, unless defined above, the transfer risks from other clauses to other variables are all zero.

Lemma 8. $\gamma \in$ *M 1 in* 3*-SAT* $\implies \beta \in$ *SS α-IDS.*

Proof. Given a satisfiable assignment for γ, we show how to construct a PSNE for β. Let $x^{(1)} = \{i \in [m] \ : \ x_i = 1\}$ be the indices of the variables that are assigned a value of one in the satisfiable assignment. For consistence, we let a_i to denote the action of any player $i \in [n]$ and construct a PSNE as follows. For each of the variable player $i \in \{2c+1, ..., 2c+m\}$, $a_i = 1$ if $(i-2c) \in x^{(1)}$ and $a_i = 0$ otherwise. Together with the partial action profile of the clauses, we will call this constructed pure-strategy profile $a = (a_1, ..., a_n)$.

To show that a is a PSNE, we argue that each player is playing its best-response. First, we consider the clause players. Recall that since the clause players are the type of SS, for each $i \in [c]$, we have

$$\mathcal{BR}_i^{ss}(a_{\text{Pa}(i)}) \equiv \begin{cases} \{0\}, & \Delta_i^{ss} < s_i(a_{\text{Pa}(i)}), \\ \{1\}, & \Delta_i^{ss} > s_i(a_{\text{Pa}(i)}), \\ \{0,1\}, & \Delta_i^{ss} = s_i(a_{\text{Pa}(i)}) \ , \end{cases}$$

where $Pa(i) = \{i_1, i_2, i_3\}$ (which corresponds to variables $x_{i_1}, x_{i_2}, x_{i_3}$ of clause i) and $s_i(a_{\text{Pa}(i)}) = \prod_{j \in \text{Pa}(i)} (1-q)^{1-a_j}$. Moreover, by the satisfiable assignment, exactly one variable in $\text{Pa}(i)$ is assigned to a value of one which corresponds to exactly one variable player that plays action one. Therefore, $s_i(a_{\text{Pa}(i)}) = (1-q)^2$. By our construction, $(1-q)^2 > \Delta_i^{ss} > (1-q)^3$. It follows that $s_i(a_{\text{Pa}(i)}) > \Delta_i^{ss}$, and the i's best-response is zero. This holds for all clause players $i \in [c]$. On the other hand, for the clause player $i \in \{c+1, ..., 2c\}$, $s_i(a_{\text{Pa}(i)}) = (1-q)^2$ as well. Since clause player i is also a SS player, we have the same best-response correspondence. By our construction, $(1-q) > \Delta_i^{ss} > (1-q)^2$, it follows that $\Delta_i^{ss} > s_i(a_{\text{Pa}(i)})$ and $a_i = 1$ is the best-response.

For each variable player $i \in \{2c+1, ..., 2c+m\}$, i has no parent and i's overall risk is 0. To determine whether i plays the action invest or not invest, we only need to compare the value of R_i and p_i. By construction, $R_i = p_i$ for all variable players i, we have that the variable players are indifferent between playing one and zero. Hence, the pure-strategy profile a is a PSNE. □

Lemma 9. $\beta \in$ *SS α-IDS* $\implies \gamma \in$ *M 1 in* 3*-SAT.*

Proof. Now we show how to construct a satisfiable assignment for γ given a PSNE of β. Let $a = (a_1, ..., a_n)$ be a PSNE of β. For each variable $i \in [m]$, if $a_{2m+i} = 1$ then $x_i = 1$ and if $a_{2m+i} = 0$ then $x_i = 0$. To show that each clause, say $i \in [c]$, has exactly one variable that is true, we observe the best-response of clause players i and $c + i$ that correspond to clause i. Given the fixed action of $a_i = 0$ and $a_{c+i} = 1$ at a PSNE, it implies that $s_i(a_{\mathrm{Pa}(i)}) > \Delta_i^{ss}$ and $s_{c+i}(a_{\mathrm{Pa}(c+i)}) < \Delta_{c+i}^{ss}$. Since $(1-q)^2 > \Delta_i^{ss} > (1-q)^3$, $(1-q) > \Delta_{c+i}^{ss} > (1-q)^2$, $\mathrm{Pa}(c+i) = \mathrm{Pa}(i)$, $|Pa(i)| = 3$, and the transfer risks are the same, we have $s_{c+i}(a_{\mathrm{Pa}(c+i)}) = (1-q)^2$. This implies that exactly one of the variables is true.

□

It is easy to see that given a (partial) pure-strategy profile, we can verify whether it is a PSNE of a SS α-IDS game in polynomial time. This fact, together with Lemma 8 and Lemma 9, we have our hardness result.

□

Combining Theorems 10 and 11 yields the next corollary.

Corollary 1. *The Pure-Nash Extension problem for n-player SC+SS α-IDS games is NP-complete.*

3.5 Preliminary Experimental Results

To illustrate the impact of the α parameter on α-IDS games, we perform experiments on randomly-generated instances of α-IDS games in which we compute a possibly approximate NE. Due to the fact that we do not have an algorithm to compute a NE in general structure α-IDS games, we will use a known heuristic to compute approximate NE.

Given $\epsilon > 0$, in an approximate ϵ-NE each individual's unilateral deviation cannot reduce the individual's expected cost by more than ϵ. More specially, in our setting, a mixed-strategy profile x is a ϵ-NE if for all players $i \in [n]$, $M_i(x) \leq M_i(1, x_{-i}) + \epsilon$ and $M_i(x) \leq M_i(0, x_{-i}) + \epsilon$.

3.5.1 Structure of the Graphs

The underlying structures of the instances use network graphs from publicly-available, real-world datasets [Zachary, 1977, Knuth, 1993, Girvan and Newman, 2002, Watts and Strogatz, 1998, Leskovec et al., 2010, Klimt and Yang, 2004]. Table 3.1 shows the exact number of nodes and edges for each of the graphs from the real-world datasets we used for our experiments.

Graph	Nodes	Edges
Karate Club	34	78
Les Miserables	77	254
College Football	115	613
Power Grid	4941	6594
Wiki Vote	7115	103689
Email Enron	36692	367662

Table 3.1: **Exact number of nodes and edges for different real-world graphs.**

Given the datasets, we view the nodes as players and the (undirected) edges as potential (bidirectional) transfer risks between the players. The number of nodes/players ranges from 34 to $\approx 37K$ while the number of edges ranges from 78 to around $368K$. The table lists the graphs in increasing size (from top to bottom).

3.5.2 Generating Parameters of α-IDS Games

To construct each instance of α-IDS games based on a given real-world datasets, for each player i, we first generate R_i where $C_i = 10^3 * (1 + \text{random}(0, 1))$ and $L_i = 10^4$ (or $L_i = 10^4/3$) to obtain a low (high) cost-to-loss ratio. The α_i value for each player i is generated independently from a normal distribution $\mathcal{N}(\mu, \sigma^2)$ with the given mean μ and standard deviation σ^2 or from a uniform distribution as specified in the experiments. After generated the R_i and α_i for each player i, we then generate p_i randomly such that such that Δ_i^{sc} or Δ_i^{ss} is in $[0, 1]$. Finally, we generate q_{ji}'s that are consistent with probabilistic constraints relative to the other parameters (i.e. $p_i + \sum_{j \in Pa(i)} q_{ji} \leq 1$).

3.5.3 Computing Approximate NE of α-IDS Games

Given a randomly generated instance of α-IDS games, we initialize the players' mixed strategies uniformly at random and run a simple gradient-dynamics heuristic based on regret minimization [Fudenberg and Levine, 1998, Nisan et al., 2007, Shoham and Leyton-Brown, 2009] until we reach an ϵ-NE. In short, we update the strategies of all non-ϵ-best-responding players i at each round t according to $x_i^{(t+1)} \leftarrow x_i^{(t)} - 10 \times (M_i(1, \mathbf{x}_{\mathrm{Pa}(i)}^{(t)}) - M_i(0, \mathbf{x}_{\mathrm{Pa}(i)}^{(t)}))$. Note that for ϵ-NE to be well-defined, all M_is' values are normalized. Given that our main interest is to study the structural properties of arbitrary α-IDS games, our hardness results of computing NE in such games justify the use of a heuristic as we do here. (Kearns and Ortiz [2004] also used a similar heuristic in their experiments.)

Given a dataset, we generated ten instances of α-IDS games and record the total percentage of level of investment, which is the sum of the mixed-strategies of the players divided by the number of players, at an approximate NE. Table 3.2 shows the average level of investment at NE on each graph instance. In the table, we consider the graph structures using the Karate Club, Les Miserables, College Football, Power grid, Wiki Voters, and Enron Email networks. The standard deviations are not shown because they are not significant. Each row represents a dataset and shows the average percentage of SS players in the game (first column), the average percentage level of investment of the SC players in the game (second column), and the average percentage level of investment of the SS players in the game (third column) of a given normal distributions or a uniform distribution for the α values

In particular, we consider α values generated from $\mathcal{N}(0.4, 0.2)$, $\mathcal{N}(0.8, 0.2)$, and uniform distribution from $[0, 1]$. Regardless of the datasets and on the instances of α-IDS games that are generated according to the process discussed earlier, we observe that as we change the distribution of α from $\mathcal{N}(0.4, 0.2)$, $\mathcal{N}(0.8, 0.2)$, there are more percentage of SC players in the system, which is consistent with the nature of the game instances. The uniform distribution yields the percentage of SC players somewhere in between the two normal distributions but closer to the percentage of $\mathcal{N}(0.4, 0.2)$. This observation holds in both high and low cost-to-loss ratios.

Moreover, in all of the instances and all of the datasets, almost all of the SC players play the action invest while the SS players play the action do not invest. This makes sense because of the nature of the SC and SS players.

Going from high to low cost-to-loss ratio, we see that the number of SS players and the percentage of SS players player the action invest at approximate NE increase across all α values.

High $\frac{C_i}{L_i}$	$\alpha_i \sim \mathcal{N}(0.4, 0.2)$			$\alpha_i \sim \mathcal{N}(0.8, 0.2)$		
Datasets	%SS	%SC Invest	%SS Invest	%SS	%SC Invest	%SS Invest
Karate Club	76.18	100.00	21.37	12.35	100.00	0.00
Les Miserables	75.45	100.00	17.93	11.82	99.85	0.67
College Football	75.65	100.00	15.47	11.57	100.00	0.00
Power Grid	75.47	97.76*	19.38*	12.82	98.79*	2.13*
Wiki Vote	75.55	97.46*	17.87*	12.78	98.92*	2.06*
Email Enron	75.29	95.97*	19.91*	12.53	97.92*	2.24*
High $\frac{C_i}{L_i}$	$\alpha_i \in [0, 1]$					
Datasets	%SS	%SC Invest	%SS Invest			
Karate Club	56.18	100.00	14.88			
Les Miserables	55.06	99.40	14.84			
College Football	55.39	100.00	13.46			
Power Grid	55.01	97.31**	15.90**			
Wiki Vote	55.02	97.00**	14.75**			
Email Enron	54.78	94.39**	16.84**			
Low $\frac{C_i}{L_i}$	$\alpha_i \sim \mathcal{N}(0.4, 0.2)$			$\alpha_i \sim \mathcal{N}(0.8, 0.2)$		
Karate Club	99.41	100.00	49.64	60.59	100.00	23.19
Les Miserables	98.96	100.00	51.17	59.22	100.00	28.34
College Football	98.87	100.00	60.42	61.48	100.00	28.30
Power Grid	98.68	99.13*	49.45*	59.41	98.81*	28.66*
Wiki Vote	98.62	98.30*	46.50*	59.89	97.38*	27.54*
Email Enron	98.73	97.96**	49.80**	59.85	96.48*	29.32*
Low $\frac{C_i}{L_i}$	$\alpha_i \in [0, 1]$					
Karate Club	86.18	100.00	41.34			
Les Miserables	85.71	100.00	49.26			
College Football	86.35	100.00	54.87			
Power Grid	85.20	99.13**	45.07**			
Wiki Vote	85.01	98.51**	44.45**			
Email Enron	84.94	98.0**	44.72**			

*=0.001-NE, **=0.005-NE, %SS (%SC) = Percentage of SS (SC) players,
$\mathcal{N}(\mu, \sigma^2)$ =normal distribution with mean μ and variance σ^2

Table 3.2: **Level of Investment of SC+SS α-IDS Games at Nash Equilibrium.**

3.6 Conclusion

In this chapter, we extend the original IDS games and introduce α-IDS games to capture the situation in which the players can partially protect transfer risks from others in the games. We partition the players into the class of SC players and SS players based on their α values and direct risks. Along with model, we provide some algorithmic and hardness results on computing NE in various classes of α-IDS games where there are only SC, SS, and SC+SS players in the games. From our results, the case of SC α-IDS games is mostly understood. However, we still do not know much about the α-IDS games that consist of SS player. In particular, it is remain open to look at the computational complexity of computing a PSNE of SS α-IDS games. Indeed, the question of computing an approximate NE has not been explored in this work. Therefore, it would be interesting to see if there an efficient algorithm to compute approximate NE in general SC+SS α-IDS games.

Chapter 4

Interdependent Defense Games[1]

Building from the generalized Interdependent Security games, in this chapter, we introduce *Interdependent Defense (IDD) games.* We begin by introducing an additional player, the *attacker*, who *deliberately* initiates bad events. (So that now bad events are no longer "chance occurrences" without any strategic deliberation.) The attacker has a *target decision* for each player - a choice of attack ($b_i = 1$) or not attack ($b_i = 0$) player i. Hence, the attacker's pure strategy is denoted by the vector $b \in \{0, 1\}^n$.

Changing from "random" non-strategic attacks whose probability of occurrence is determined independent of the actions of the internal players, to intentional attacks, ones that are deliberately carried out by an external actor, gives reason for us to alter p_i and q_{ij} because their original definitions actually imply extra meaning with respect to the new aggressor.

The game parameter p_i implicitly "encodes" b_i because $b_i = 0$ implies $p_i = 0$. Thus, we redefine

$$p_i \equiv p_i(b_i) \equiv b_i \widehat{p}_i$$

so that player i has *intrinsic risk* $\widehat{p}_i$, and only has *internal risk* if targeted (i.e, $b_i = 1$). The new parameter $\widehat{p}_i$ represents the (*conditional*) probability that an attack is successful at player i *given* that player i was directly targeted and did not invest in protection.

[1] A part of this chapter has appeared in the proceedings of the *Twenty-eighth Conference on Uncertainty in Artificial Intelligence (UAI 2012).*

The game parameter q_{ij} "encodes" $b_i = 1$, because a prerequisite is that i is targeted before it can transfer the bad event to j. We redefine

$$q_{ij} \equiv q_{ij}(b_i) \equiv b_i \widehat{q}_{ij}$$

so that $\widehat{q}_{ij}$ is the intrinsic transfer probability from player i to player j, independent of b_i. The new parameter $\widehat{q}_{ij}$ represents the (*conditional*) probability that an attack is successful at player j *given* that it originated at player i, did not occur at i but was transferred undetected to j. Note that just as it was the case with traditional IDS games, there is an implicit constraint on the risk-related parameters: $\widehat{p}_i + \sum_{j \in \mathrm{Pa}(i)} \widehat{q}_{ji} \leq 1$, for all i.

Because the p_i's and q_{ij}'s depend on the attacker's action $\mathbf{b}$, so does the safety and risk functions. In particular, we now have

$$e_{ij}(a_j, b_j) \equiv a_j + (1 - a_j)(1 - b_j \widehat{q}_{ji}) = (1 - \widehat{q}_{ji})^{b_j(1-a_j)},$$

$s_i(a_{\mathrm{Pa}(i)}, b_{\mathrm{Pa}(i)}) \equiv \prod_{j \in \mathrm{Pa}(i)} e_{ij}(a_j, b_j) \equiv 1 - r_i(a_{\mathrm{Pa}(i)}, b_{\mathrm{Pa}(i)})$. Hence, for each player i, the *cost* function becomes

$$\begin{aligned} M_i(a_i, a_{\mathrm{Pa}(i)}, b_i, b_{\mathrm{Pa}(i)}) \equiv{} & a_i[C_i + \alpha_i r_i(a_{\mathrm{Pa}(i)}, b_{\mathrm{Pa}(i)}) L_i] \\ & + (1 - a_i)[b_i \widehat{p}_i + (1 - b_i \widehat{p}_i) r_i(a_{\mathrm{Pa}(i)}, b_{\mathrm{Pa}(i)})] L_i. \end{aligned}$$

We assume the attacker wants to cause as much damage as possible. One possible *utility/payoff* function U quantifying the objective of the attacker is

$$U(a, b) \equiv \textstyle\sum_{i=1}^{n} M_i(a_{\mathrm{PF}(i)}, b_{\mathrm{PF}(i)}) - a_i C_i - b_i C_i^0.$$

which adds the expected players costs (for targeted and transferred bad events) over all players, minus C_i^0, the attacker's own *"cost" to target player i*.

Of course, many other utility functions of varied complexity are also possible. Indeed, one can consider increasingly complex and sophisticated utility functions that may explicitly parse out the involved costs and induced losses in finer-grain and painstaking detail. For instance, we could decompose the cost to the attacker to target a specific site into different components such as, perhaps, planning and setup costs, carry-out cost, the costs of getting caught or retaliated against, etc. We leave these more complex variants for future work.

We close out this section by presenting the attacker's *best-response correspondence* $\mathcal{BR}_0 : \{0,1\}^n \to 2^{\{0,1\}^n}$:

$$\mathcal{BR}_0(a) \equiv \arg\max_{\mathbf{b} \in \{0,1\}^n} U(a, b) \ . \tag{4.1}$$

Definition 12. *A pure-strategy profile* $(a^*, b^*) \in \{0,1\}^{2n}$ *is a PSNE of an IDD game if, for each player* i, $a_i^* \in \mathcal{BR}_i(a^*_{Pa(i)}, b^*_{PF(i)})$, *and for the aggressor,* $b^* \in \mathcal{BR}_0(a^*)$.

4.1 Mixed Strategies in IDD Games

For all player i, denote by x_i the *mixed strategy of player* i: the probability that player i invests. Similarly, y denotes the joint probability mass function (PMF) corresponding to the *attacker's mixed strategy* so that for all $b \in \{0,1\}^n$, $y(b)$ is the probability that the attacker executes joint-attack vector b.

Denote the marginal PMF over a subset $I \subset [n]$ of the internal players by y_I such that for all b_I, $y_I(b_I) \equiv \sum_{b_{-I}} y(b_I, b_{-I})$ is the (marginal) probability that the attacker chooses a joint-attack vector in which the sub-component decisions corresponding to players in I are as in b_I.

Denote simply by $y_i \equiv y_{\{i\}}(1)$ the marginal probability that the attacker chooses an attack vector in which player i is directly targeted.

Slightly abusing notation, we redefine the function e_{ij} (i.e., how safe i is from j), s_i and r_i (i.e., the overall transfer safety and risk, respectively) as

$$e_{ij}(x_j, b_j) \equiv x_j + (1 - x_j)(1 - b_j \widehat{q}_{ji}),$$

$$s_i(x_{\mathrm{Pa}(i)}, b_{\mathrm{Pa}(i)}) \equiv \textstyle\prod_{j \in \mathrm{Pa}(i)} e_{ij}(x_j, b_j),$$

$$s_i(x_{\mathrm{Pa}(i)}, y_{\mathrm{Pa}(i)}) \equiv \sum_{b_{\mathrm{Pa}(i)}} y_{\mathrm{Pa}(i)}(b_{\mathrm{Pa}(i)}) s_i(x_{\mathrm{Pa}(i)}, b_{\mathrm{Pa}(i)}) \ ,$$

and $r_i(x_{\mathrm{Pa}(i)}, y_{\mathrm{Pa}(i)}) \equiv 1 - s_i(x_{\mathrm{Pa}(i)}, y_{\mathrm{Pa}(i)})$.

In general, the *expected* cost of protection to site i, with respect to a mixed-strategy profile (x, y), can be expressed as

$$\begin{aligned} M_i(x_i, x_{\mathrm{Pa}(i)}, y_{\mathrm{PF}(i)}) \equiv {} & x_i[C_i + \alpha_i r_i(x_{\mathrm{Pa}(i)}, y_{\mathrm{Pa}(i)}) L_i] + \\ & (1 - x_i)[\widehat{p}_i f_i(x_{\mathrm{Pa}(i)}, y_{\mathrm{PF}(i)}) + r_i(x_{\mathrm{Pa}(i)}, y_{\mathrm{Pa}(i)})] L_i \ , \end{aligned}$$

where $f_i(x_{\mathrm{Pa}(i)}, y_{\mathrm{PF}(i)}) \equiv$

$$\textstyle\sum_{b_{\mathrm{PF}(i)}} y_{\mathrm{PF}(i)}(b_{\mathrm{PF}(i)}) \ b_i s_i(x_{\mathrm{Pa}(i)}, b_{\mathrm{Pa}(i)}) \ .$$

The expected payoff of the attacker is

$$U(x, y) \equiv \textstyle\sum_{i=1}^n M_i(x_{\mathrm{PF}(i)}, y_{\mathrm{PF}(i)}) - x_i C_i - y_i C_i^0.$$

Let $\widehat{\Delta}_i \equiv \frac{C_i}{L_i \widehat{p}_i}$ and $\widehat{s}_i(x_{\mathrm{Pa}(i)}, y_{\mathrm{PF}(i)}) \equiv f_i(x_{\mathrm{Pa}(i)}, y_{\mathrm{PF}(i)}) + \frac{1-\alpha_i}{\widehat{p}_i} r_i(x_{\mathrm{Pa}(i)}, y_{\mathrm{Pa}(i)})$. The best-response correspondence of defender i is then

$$\mathcal{BR}_i(x_{\mathrm{Pa}(i)}, y_{\mathrm{PF}(i)}) \equiv \begin{cases} \{1\}, & \text{if } \widehat{s}_i(x_{\mathrm{Pa}(i)}, y_{\mathrm{PF}(i)}) > \widehat{\Delta}_i, \\ \{0\}, & \text{if } \widehat{s}_i(x_{\mathrm{Pa}(i)}, y_{\mathrm{PF}(i)}) < \widehat{\Delta}_i, \\ [0,1], & \text{if } \widehat{s}_i(x_{\mathrm{Pa}(i)}, y_{\mathrm{PF}(i)}) = \widehat{\Delta}_i. \end{cases}$$

The best-response correspondence for the attacker is simply
$\mathcal{BR}_0(x) \equiv \arg\max_y U(x, y)$.

Definition 13. *A mixed-strategy profile (x^*, y^*) is a MSNE of an IDD game if (1) for all $i \in [n]$, $x_i^* \in \mathcal{BR}_i(x^*_{Pa(i)}, y^*_{PF(i)})$ and (2) $y^* \in \mathcal{BR}_0(x^*)$.*

4.2 Model Assumptions

Note that the attacker has in principle an *exponential* number of pure strategies! This affords the attacker unrealistic amount of power. Hence, we need restriction on the attacker's power. The simplest way is to allow at most a single simultaneous attack. We can weaken this assumption to allow the attacker at most K simultaneous attacks. Even then, the number of pure strategies will grow *exponentially* in the number of potential attacks, which still renders the attacker's pure-strategy space unrealistic, especially on a very large network with twenty-thousand nodes. Worst-case, we need to consider up to 2^n number of pure strategies for K attacks as K goes to n.

Assumption 1. *The set of pure strategies of the attacker is*

$$\mathcal{B} = \left\{ b \in \{0,1\}^n \;\middle|\; \textstyle\sum_{i=1}^n b_i \leq 1 \right\}.$$

Although the above assumption is stated in terms of pure-strategies, it can be easily extended to the mixed-strategy setting where we require $\sum_{i=0}^n y_i \leq 1$ and y_0 is the probability of no attack.

The following assumptions are on the game *parameters*. The next assumption states that every site's investment cost is positive and (strictly) smaller than the *conditional* expected *direct* loss if the site were to be attacked directly ($b_i = 1$); that is, if a site knows that an attack is directed against it, the site will prefer to invest in security, unless the *external risk* is too high. This assumption is reasonable because otherwise the player will never invest regardless of what other players do (i.e., not investing would be a dominant strategy).

Assumption 2. *For all sites $i \in [n]$, $0 < C_i < \widehat{p}_i L_i$.*

The next assumption states that, for all sites i, the attacker's cost to attack i is positive and (strictly) smaller than the expected loss (i.e., gains from the perspective of the attacker) achieved if an attack initiated at site i is successful, either directly at i or at one of its children (after transfer); that is, if an attacker knows that an attack is rewarding (or able to obtain a positive utility), it will prefer to attack some nodes in the network. This assumption is reasonable; otherwise the attacker will never attack regardless of what other players do (i.e., not attacking would be a dominant strategy, leading to an easy problem to solve).

Assumption 3. *For all sites $i \in [n]$, $0 < C_i^0 < \widehat{p}_i L_i + \sum_{j \in Ch(i)} \widehat{q}_{ij} \alpha_j L_j$.*

In the following, we will study the problem of finding and computing NE in IDD games under the above three assumptions.

4.3 PSNE of IDD Games

It turns out that under these three assumptions, there is no PSNE in IDD games. This is typical of attacker-defender settings. The following proposition eliminates PSNE as a universal solution concept for natural IDD games in which at most one attack is possible. The main significance of this result is that it allows us to concentrate our efforts on the much harder problem of computing MSNE.

Proposition 3. *No IDD game in which Assumptions 1, 2 and 3 hold has a PSNE.*

Proof. First note that Assumption 1 considerably simplifies some of the expressions involving external risk/safety. This is because any pure strategy in $\mathcal{B}$ is either a vector of all 0's, or exactly one 1. For instance, in this case we have

$$\begin{aligned} s_i(a_{\mathrm{Pa}(i)}, b_{\mathrm{Pa}(i)}) &= \begin{cases} \sum_{j \in \mathrm{Pa}(i)} b_j e_{ij}(a_j, 1), & \text{if } b_k = 1 \text{ for some } k \in \mathrm{Pa}(i), \\ 1, & \text{if } b_k = 0 \text{ for all } k \in \mathrm{Pa}(i), \end{cases} \\ &= 1 - \sum_{j \in \mathrm{Pa}(i)} b_j (1 - a_j) \widehat{q}_{ji}, \end{aligned}$$

so that

$$r_i(a_{\mathrm{Pa}(i)}, b_{\mathrm{Pa}(i)}) = \sum_{j \in \mathrm{Pa}(i)} b_j(1 - a_j)\widehat{q}_{ji},$$

and

$$b_i\, s_i(a_{\mathrm{Pa}(i)}, b_{\mathrm{Pa}(i)}) = b_i.$$

Also, if the IDD game has a PSNE (a^*, b^*), then the attacker's payoff in it is

$$U(a^*, b^*) = \left[\max_{i \in [n]} (1 - a_i^*) \left(\widehat{p}_i L_i + \sum_{j \in \mathrm{Ch}(i)} \widehat{q}_{ij}(a_j^* \alpha_j + (1 - a_j^*))L_j\right) - C_i^0\right]^+$$

where for any real number $z \in \mathbb{R}$, the operator $[z]^+ \equiv \max(z, 0)$; in addition, if $b_k^* = 1$ for some $k \in [n]$, then

$$(1 - a_k^*) \left(\widehat{p}_k L_k + \sum_{j \in \mathrm{Ch}(k)} \widehat{q}_{kj}(a_j^* \alpha_j + (1 - a_j^*))L_j\right) - C_k^0 \geq$$
$$\left[\max_{i \in [n]} (1 - a_i^*) \left(\widehat{p}_i L_i + \sum_{j \in \mathrm{Ch}(i)} \widehat{q}_{ij}(a_j^* \alpha_j + (1 - a_j^*))L_j\right) - C_i^0\right]^+ . \qquad (4.2)$$

The proof of the proposition is by contradiction. Consider an IDD game that satisfies the conditions of the proposition. Let (a^*, b^*) be a PSNE of the game. We need to consider two cases at the PSNE: (1) there is some attack and (2) there is no attack.

1. If there is some attack, then $b_k^* = 1$ for some site $k \in [n]$, and for all $i \neq k$, $b_i^* = 0$. In addition, because b^* is consistent with the aggressor's best response to a^*, we have, using condition 4.2 above,

   $$(1 - a_k^*) \left(\widehat{p}_k L_k + \sum_{j \in \mathrm{Ch}(k)} \widehat{q}_{kj}(a_j^* \alpha_j + (1 - a_j^*))L_j\right) \geq C_k^0 > 0 ,$$

   The last condition and Assumption 3 implies $a_k^* = 0$. Hence, by the best-response condition of site k, we have

   $$C_k + \alpha_k r_k(a^*_{\mathrm{Pa}(k)}, b^*_{\mathrm{Pa}(k)})L_k \geq \widehat{p}_k L_k + (1 - \widehat{p}_k) r_k(a^*_{\mathrm{Pa}(k)}, b^*_{\mathrm{Pa}(k)})L_k .$$

Because the attack occurs at k, the transfer risk $r_k(a^*_{\mathrm{Pa}(k)}, b^*_{\mathrm{Pa}(k)}) = r_k(a^*_{\mathrm{Pa}(k)}, \mathbf{0}) = 0$ at the PSNE. Therefore, the last condition simplifies to

$$C_k \geq \widehat{p}_k L_k \ ,$$

which contradicts Assumption 2.

2. If there is no attack, then $b^* = \mathbf{0}$. In this case, the site's best-response conditions imply $a^* = \mathbf{0}$. From the attacker's best-response condition we obtain

$$\widehat{p}_k L_k + \sum_{j \in \mathrm{Ch}(k)} \widehat{q}_{kj} L_j \leq C^0_k \ ,$$

which contradicts Assumption 3.

□

Now, we will concentrate our efforts on the much harder problem of computing MSNE.

4.4 MSNE of IDD Games

We first consider the IDD games where the players' investments cannot reduce the overall risk. This is the same setting in the original IDS games.

Assumption 4. *For all internal players $i \in N$, the probability that player i's investment in security does not protect the player from transfers, α_i, is* 1.

For convenient, we will characterize the type of IDD games based on the imposed assumptions.

Definition 14. *We say an IDD game is* transfer-vulnerable *if Assumption 4 holds. We say an IDD game is a* single-simultaneous-attack *game if Assumption 1 holds (i.e., at most one attack is possible).*

Assumption 1, in the context of mixed strategies, implies the probability of no attack $y_0 \equiv 1 - \sum_i^n y_i$. Assumptions 1 and 4 greatly simplify the best-response condition of the internal players because now $\widehat{s}_i(\mathbf{x}_{\mathrm{Pa}(i)}, y_{\mathrm{PF}(i)}) = y_i$.

Let $L^0_i(x_i) \equiv (1 - x_i)(\widehat{p}_i L_i + \sum_{j \in \mathrm{Ch}(i)} \widehat{q}_{ij} L_j)$. It will also be convenient to denote by $\bar{L}^0_i \equiv L^0_i(0) = \widehat{p}_i L_i + \sum_{j \in \mathrm{Ch}(i)} \widehat{q}_{ij} L_j$, so that we can express $L^0_i(x_i) = (1 - x_i)\bar{L}^0_i$, to highlight that L^0_i is a linear function of x_i.

Similarly, it will also be convenient to let $M_i^0(x_i) \equiv L_i^0(x_i) - C_i^0$, and denote $\bar{M}_i^0 \equiv M_i^0(0) = \bar{L}_i^0 - C_i^0$. Let $\eta_i^0 \equiv C_i^0/\bar{L}_i^0$. The best-response condition of the attacker also simplifies under the same assumptions because now $U(x,y) = \sum_{i=1}^n y_i M_i^0(x_i)$.

Assumption 3 is reasonable in our new context because, under Assumption 4, if there were a player i with $\eta_i^0 > 1$, the attacker would never attack i, and as a result player i would never invest. In that case, we can safely remove j from the game, without any loss of generality.

We now characterize the space of MSNE in IDD games, which will immediately lead to a polynomial-time algorithm for computing *all* MSNE.

4.4.1 Characterization

The characterization starts by partitioning the space of games into three, based on whether $\sum_{i=1}^n \widehat{\Delta}_i$ is (1) <, (2) =, or (3) > than 1. The rationale behind this is that now the players are indifferent between investing or not investing when $y_i = \widehat{\Delta}_i$, by the best-response correspondence the attacker's mixed strategy is restricted. The following result fully characterizes the set of MSNE in single simultaneous attack transfer-vulnerable IDD games.

Proposition 4. *The mixed-strategy profile (x^*, y^*) is an MSNE of a single-simultaneous-attack transfer-vulnerable IDD game in which*

1. $\sum_{i=1}^n \widehat{\Delta}_i < 1$ *if and only if (1)* $1 > y_0^* = 1 - \sum_{i=1}^n \widehat{\Delta}_i > 0$*, and (2) for all i,* $y_i^* = \widehat{\Delta}_i > 0$ *and* $0 < x_i^* = 1 - \eta_i^0 < 1$.

2. $\sum_{i=1}^n \widehat{\Delta}_i = 1$ *if and only if (1)* $y_0^* = 0$*, and (2) for all i,* $y_i^* = \widehat{\Delta}_i > 0$ *and* $x_i^* = 1 - \frac{v + C_i^0}{\bar{L}_i^0}$ *with* $0 \leq v \leq \min_{i \in [n]} \bar{M}_i^0$.

3. $\sum_{i=1}^n \widehat{\Delta}_i > 1$ *if and only if (1)* $y_0^* = 0$*, and (2) there exists a non-singleton, non-empty subset* $I \subset [n]$*, such that* $\min_{i \in I} \bar{M}_i^0 \geq \max_{k \notin I} \bar{M}_k^0$ *if* $I \neq [n]$*, and the following holds: (a) for all* $k \notin I$, $x_k^* = 0$ *and* $y_k^* = 0$*, (b) for all* $i \in J \equiv \arg\min_{i \in I} \bar{M}_i^0$, $x_i^* = 0$ *and* $0 \leq y_i^* \leq \widehat{\Delta}_i$*, and in addition,* $\sum_{i \in J} y_i^* = 1 - \sum_{t \in I - J} \widehat{\Delta}_i$*; and (c) for all* $i \in I - J$, $y_i^* = \widehat{\Delta}_i$ *and* $0 < x_i^* = 1 - \frac{\min_{t \in I} \bar{M}_t^0 + C_i^0}{\bar{L}_i^0} < 1$.

As proof sketch, we briefly state that the proposition follows from the restrictions imposed by the model parameters and their implication to indifference and monotonicity conditions. We also mention that the third case in the proposition implies that if the $\bar{M}_l^0$'s form a complete order, then the last condition stated in that case allows us to search for a MSNE by exploring only $n-2$ sets, vs. 2^{n-2} if done naively.

It turns out a complete order is not necessary. The following claim allows us to safely move all the internal players with the same value of $\bar{M}_i^0$ in a group as a whole inside or outside I.

Claim 4. *Let $I \subset [n]$, such that $I' \subset I$, $|I'| < |I| < n-1$. Suppose we find an MSNE (x, y) such that $I' = \{i \mid y_i > 0\}$, with the property that $\min_{l \in I'} \bar{M}_l^0 = \max_{k \notin I'} \bar{M}_k^0$. In addition, suppose I satisfies $\min_{l \in I'} \bar{M}_l^0 = \min_{l \in I} \bar{M}_l^0 \geq \max_{k \notin I} \bar{M}_k^0$. Then, we can also find (x, y) using the partition imposed by I.*

Proof. To simplify the notation, let $v \equiv \min_{l \in I} \bar{M}_l^0 = \min_{l \in I'} \bar{M}_l^0$, $J' \equiv \arg\min_{l \in I'} \bar{M}_l^0$ and $J \equiv \arg\min_{i \in I} \bar{M}_i^0$. The hypothesis implies that $(\mathbf{x}, \mathbf{y})$ satisfies the following properties.

$$\text{for all } i \notin I'\text{: } x_i = y_i = 0$$

$$\text{for all } i \in J'\text{: } x_i = 0 \text{ and } 0 \leq y_i \leq \widehat{\Delta}_i; \text{ also } \sum_{i \in J'} y_i = 1 - \sum_{i \in I' - J'} \widehat{\Delta}_i$$

$$\text{for all } i \in I' - J'\text{: } x_i = 1 - \frac{v + C_i^0}{\bar{L}_i^0} \text{ and } y_i = \widehat{\Delta}_i$$

We now show that (x, y) also satisfies the constraints when using I with the properties stated in the claim. For that, it needs to satisfy the same expressions as above, but with I' and J' replaced by I and J, respectively.

The first condition is satisfied because $I' \subset I$. The second condition is satisfied for all $i \in J - I'$, because $i \notin I'$ satisfies $x_i = 0$ and $0 \leq y_i = 0 \leq \widehat{\Delta}_i$. It is also satisfied for all $i \in J \cap I'$ because $i \in J$ implies $\bar{M}_i^0 = v$ and, because $i \in I'$, $i \in J'$. For the third condition, note that $I - J \subset I' - J'$ because $i \in I - J$ implies the inequality $\bar{M}_i^0 > v = \max_{k \notin I'} \bar{M}_k^0$; hence, the first inequality in the last expression implies $i \notin J'$, while the equality implies $i \in I'$. □

Proof of Proposition 4

Throughout this proof, by the hypothesis of the proposition, we assume we are dealing with single-simultaneous-attack transfer-vulnerable IDD games.

We also use the same notation as that introduced before the statement of the proposition in the main text.

First recall that Assumption 1, in the context of mixed strategies, implies the probability of no attack $y_0 \equiv 1 - \sum_i^n y_i$. This is because under this assumption

$$y(b) = \begin{cases} y_{B_i}(b_i) = y_i, & \text{if } b_i = 1 \text{ for } \textit{exactly one } i \in [n], \\ y_0, & \text{if } b_i = 0 \text{ for } \textit{all } i \in [n], \\ 0, & \text{otherwise.} \end{cases}$$

Recall also that, when used in combination, Assumptions 1 and 4 greatly simplify the best-response condition of the internal players because now $\widehat{s}_i(x_{\mathrm{Pa}(i)}, y_{\mathrm{PF}(i)}) = y_i$. In particular, we have [2]

$$\begin{aligned} s_i(x_{\mathrm{Pa}(i)}, y_{\mathrm{Pa}(i)}) &\equiv \sum_{b_{\mathrm{Pa}(i)}} y_{\mathrm{Pa}(i)}(b_{\mathrm{Pa}(i)}) s_i(x_{\mathrm{Pa}(i)}, b_{\mathrm{Pa}(i)}) \\ &= \sum_{b_{\mathrm{Pa}(i)}} y_{\mathrm{Pa}(i)}(b_{\mathrm{Pa}(i)}) \prod_{j \in \mathrm{Pa}(i)} e_{ij}(x_j, b_j) \\ &= \left(y_0 + \sum_{j \in [n] - \mathrm{Pa}(i)} y_j \right) + \sum_{j \in \mathrm{Pa}(i)} y_j e_{ij}(x_j, 1) \\ &= \left(y_0 + \sum_{j \in [n] - \mathrm{Pa}(i)} y_j \right) + \sum_{j \in \mathrm{Pa}(i)} y_j (x_j + (1 - x_j)(1 - \widehat{q}_{ji})) \\ &= \left(y_0 + \sum_{j \in [n] - \mathrm{Pa}(i)} y_j \right) + \sum_{j \in \mathrm{Pa}(i)} y_j (x_j + (1 - x_j) - (1 - x_j)\widehat{q}_{ji}) \\ &= \left(y_0 + \sum_{j \in [n] - \mathrm{Pa}(i)} y_j \right) + \sum_{j \in \mathrm{Pa}(i)} y_j (1 - (1 - x_j)\widehat{q}_{ji}) \\ &= \left(y_0 + \sum_{j \in [n] - \mathrm{Pa}(i)} y_j \right) + \sum_{j \in \mathrm{Pa}(i)} y_j - \sum_{j \in \mathrm{Pa}(i)} y_j (1 - x_j)\widehat{q}_{ji} \\ &= 1 - \sum_{j \in \mathrm{Pa}(i)} y_j (1 - x_j)\widehat{q}_{ji} \ , \end{aligned}$$

[2] Note that $e_{ij}(x_j, 0) = 1$.

so that $r_i(x_{\mathrm{Pa}(i)}, y_{\mathrm{Pa}(i)}) = \sum_{j \in \mathrm{Pa}(i)} y_j(1-x_j)\widehat{q}_{ji}$, and

$$\begin{aligned}
f_i(x_{\mathrm{Pa}(i)}, y_{\mathrm{Pa}(i)}) &\equiv \sum_{b_{\mathrm{PF}(i)}} y_{\mathrm{PF}(i)}(b_{\mathrm{PF}(i)})\ b_i\ s_i(x_{\mathrm{Pa}(i)}, b_{\mathrm{Pa}(i)}) \\
&= \sum_{b_{\mathrm{PF}(i)}} y_{\mathrm{PF}(i)}(b_{\mathrm{PF}(i)})\ b_i \prod_{j \in \mathrm{Pa}(i)} e_{ij}(x_j, b_j) \\
&= \left(y_0 + \sum_{j \in [n]-\mathrm{PF}(i)} y_j \right) \times 0 \times 1 + y_i + \sum_{j \in \mathrm{Pa}(i)} y_j \times 0 \times e_{ij}(x_j, 1) \\
&= y_i \ .
\end{aligned}$$

Combining the last derivation above with Assumption 4 (i.e., $\alpha_i = 1$) leads to

$$\widehat{s}_i(x_{\mathrm{Pa}(i)}, y_{\mathrm{PF}(i)}) \equiv f_i(x_{\mathrm{Pa}(i)}, y_{\mathrm{PF}(i)}) + \frac{1-\alpha_i}{\widehat{p}_i} r_i(x_{\mathrm{Pa}(i)}, y_{\mathrm{Pa}(i)}) = y_i \ ,$$

as claimed above. Hence, the best-response $\mathcal{BR}_i$ of defender i *directly* depends on y_i *only* (i.e., $\mathcal{BR}_i$ is *conditionally* independent of the mixed-strategies $x_{\mathrm{Pa}(i)}$ of its parent nodes $\mathrm{Pa}(i)$ of defender node i in the network *given* the probability y_i that the attacker's mixed-strategy y assigns to a direct attack to i); thus, in what follows, we abuse notation and define

$$\mathcal{BR}_i(y_i) \equiv \mathcal{BR}_i(x_{\mathrm{Pa}(i)}, y_{\mathrm{PF}(i)}) = \begin{cases} \{1\}, & \text{if } y_i > \widehat{\Delta}_i, \\ \{0\}, & \text{if } y_i < \widehat{\Delta}_i, \\ [0,1], & \text{if } y_i = \widehat{\Delta}_i. \end{cases}$$

Next, we prove some useful properties of the MSNE.

Claim 5. *In every MSNE (x, y), for all $i \in [n]$, if the probability of a direct attack to a defender i is $y_i = 0$ then the probability of investment of defender i is $x_i = 0$. In addition, if $y_i = 0$ for some $i \in [n]$ then the probability of no attack $y_0 = 0$.*

Proof. By $\mathcal{BR}_i$, $y_i = 0 < \widehat{\Delta}_i$ implies $x_i = 0$. For the second part, if $y_i = 0$ for some defender $i \in [n]$, then, by $\mathcal{BR}_0$, we have

$$\max_t M_t^0(x_t) \geq M_i^0(x_i) = \bar{M}_k^0 > 0,$$

and thus $y_0 = 0$. □

Proposition 5. *In every MSNE (x, y), an attack is always possible: $y_0 < 1$.*

Proof. The proof is by contradiction. Let (x, y) be an MSNE. Suppose there is no attack: $y_0 = 1$. Then, $\sum_{i=1}^{n} y_i = 1 - y_0 = 0$, so that $y_i = 0$ for all $i \in [n]$. Because $y_i = 0$ for some $i \in [n]$, Claim 5 yields $y_0 = 0$, a contradiction. □

Lemma 10. *In every MSNE (x, y), the probability y_i of direct attack to defender i is no larger than $\widehat{\Delta}_i < 1$.*

Proof. The proof is by contradiction. Suppose there is some MSNE in which $y_i > \widehat{\Delta}_i$ for some $i \in [n]$. Then, $x_i = 1$ and in turn $M_i^0(1) = -C_i^0 < 0$. Because the attacker can always achieve expected payoff 0 by not attacking anyone, the last condition implies $y_i = 0$, a contradiction. □

Claim 6. *Let y be the mixed-strategy of the attacker in some MSNE. If the probability of no attack $y_0 > 0$, then the probability of direct attack to defender i is equal to the cost-to-conditional expected-loss of defender i: $y_i = \widehat{\Delta}_i$ for all $i \in [n]$.*

Proof. The proof is by contradiction. By Lemma 10 $y_i \leq \widehat{\Delta}_i$ for all $i \in [n]$. Suppose $y_i < \widehat{\Delta}_i$ for some i. Then, by $\mathcal{BR}_i$, we have $x_i = 0$, and by $\mathcal{BR}_0$, we have $0 \geq \bar{M}_i^0 > 0$, a contradiction. □

Lemma 11. *In every MSNE (x, y) of an IDD game in which the total of cost-to-conditional expected-loss of all defenders is $\sum_{i=1}^{n} \widehat{\Delta}_i < 1$, there may not be an attack: $y_0 > 0$.*

Proof. By Lemma 10, $y_i \leq \widehat{\Delta}_i$ for all $i \in [n]$. Using the last statement, note that

$$1 - y_0 = \sum_{i=1}^{n} y_i \leq \sum_{i=1}^{n} \widehat{\Delta}_i < 1,$$

from which the lemma immediately follows. □

As stated earlier, we partition the class of IDD games into three subclasses, based on whether $\sum_{i=1}^{n} \widehat{\Delta}_i$ is (1) less than, (2) equal to, or (3) greater than 1. We consider each subclass in turn.

Proposition 6. *The mixed-strategy profile (x, y) is an MSNE of an IDD game in which the total cost-to-conditional expected-loss of all defenders is $\sum_{i=1}^{n} \widehat{\Delta}_i < 1$ if and only if it satisfies the following properties.*

1. *There may not be an attack, and the probability of no attack is equal to one minus the cost-to-conditional expected-loss of all defenders:* $1 > y_0 = 1 - \sum_{i=1}^{n} \widehat{\Delta}_i > 0$.

2. *Every defender has non-zero chance of being attacked directly, and this probability equals the respective defender's cost-to-conditional expected-loss of defender: for all defenders* $i \in [n]$*,* $y_i = \widehat{\Delta}_i > 0$.

3. *Every defender invests some but none does fully, and in particular, the probability a defender does* not *invest equals the respective cost-to-loss ratio to the attacker: for all defenders* $i \in [n]$*,* $0 < x_i = 1 - \eta_i^0 < 1$.

Proof. Suppose the mixed-strategy profile (x, y) satisfies the above properties. Then, every defender is indifferent (i.e., for all $i \in [n]$, $\mathcal{BR}_i(y_i) = [0, 1]$, because $y_i = \widehat{\Delta}_i$), as is also the attacker (i.e., $\mathcal{BR}_0(x)$ equals the set of *all* probability distributions over $n+1$ events because $M_i^0(x_i) = 0$ for all $i \in [n]$). Hence, (x, y) is an MSNE.

Now suppose (x, y) is an MSNE of the game. By Lemma 11, $y_0 > 0$. Hence, for all $i \in [n]$, we have $y_i = \widehat{\Delta}_i > 0$ by Claim 6. Both of the previous sentences together imply $M_i^0(x_i) = 0$ for all $i \in [n]$, because of $\mathcal{BR}_0$. Simple algebra yields that $x_i = 1 - \eta_i^0$. Finally, because $y_0 + \sum_{i=1}^{n} y_i = 1$, we have $y_0 = 1 - \sum_{i=1}^{n} \widehat{\Delta}_i$. □

Proposition 7. *The mixed-strategy profile* (x, y) *is an MSNE of an IDD game in which* $\sum_{i=1}^{n} \widehat{\Delta}_i = 1$ *if and only if it satisfies the following properties.*

1. *There is always an attack:* $y_0 = 0$.

2. *Every defender has non-zero chance of being attacked directly, and this probability equals the respective defender's cost-to-conditional expected-loss of defender i: for all defenders* $i \in [n]$*,* $y_i = \widehat{\Delta}_i > 0$.

3. *No defender invests fully, and the possible investment probabilities are connected by a 1-d line segment in* $\mathbb{R}^n$*:*

$$x_i = 1 - \frac{v + C_i^0}{\bar{L}_i^0} \textit{ for all } i \in [n]$$

with $0 \le v \le \min_{i \in [n]} \bar{M}_i^0$.

Proof. Suppose the mixed-strategy profile (x, y) satisfies the properties above. Then, every defender is indifferent: for all $i \in [n]$, $\mathcal{BR}_i(y_i) = [0, 1]$, because $y_i = \widehat{\Delta}_i$. To test $y \in \mathcal{BR}_0(x)$, note $0 \leq (1 - x_i)\bar{L}_i^0 - C_i^0 = M_i^0(x_i) = \max_{t \in [n]} M_t^0(x_t)$ for all $i \in [n]$, and

$$\sum_{i=1}^{n} y_i M_i^0(x_i) = \sum_{i=1}^{n} y_i \max_{t \in [n]} M_t^0(x_t) =$$

$$\left(\sum_{i=1}^{n} y_i\right) \max_{t \in [n]} M_t^0(x_t) = \max_{t \in [n]} M_t^0(x_t).$$

Let the mixed-strategy profile (x, y) be an MSNE of the game. Let $I \equiv I(y) \equiv \{i \in [n] \mid y_i > 0\}$. Note that $y_k = 0$ for all $k \notin I$. We first prove the following lemma.

Lemma 12. *$I = [n]$.*

Proof. The proof is by contradiction. Suppose $I \neq [n]$. By Proposition 5, $y_0 < 1 = y_0 + \sum_{i=1}^{n} y_i$ so that $y_i > 0$ for some $i \in [n]$, and therefore $I \neq \emptyset$. Also, there exists some $k \in [n] - I$, for which $y_k = 0$. By Claim 5, we then have for all $k \notin I$, $x_k = 0$. By $\mathcal{BR}_0$ and Assumption 3, for all $i, t \in I \neq \emptyset$ and $k \notin I$,

$$M_i^0(x_i) = M_t^0(x_t) \geq \bar{M}_k^0.$$

The condition above yields the following upper bound on the mixed strategies of the defenders in $i \in I$, after applying simple algebraic manipulations: for all $i \in I, k \notin I$,

$$x_i \leq 1 - \frac{\bar{M}_k^0 + C_i^0}{\bar{L}_i^0} < 1.$$

By $\mathcal{BR}_i$, this implies that $y_i \leq \widehat{\Delta}_i$ for all $i \in I$. Putting all of the above together, we have

$$1 = \sum_{i=0}^{n} y_i = \sum_{i=1}^{n} y_i = \sum_{i \in I} y_i \leq \sum_{i \in I} \widehat{\Delta}_i \leq \sum_{i=1}^{n} \widehat{\Delta}_i = 1.$$

Now, because $I \neq [n]$ (by the hypothesis assumed to obtain a contradiction), we have $\sum_{k \notin I} \widehat{\Delta}_k > 0$, and

$$\sum_{i \in I} y_i = \sum_{i=1}^{n} \widehat{\Delta}_i = \sum_{i \in I} \widehat{\Delta}_i + \sum_{k \notin I} \widehat{\Delta}_k > \sum_{i \in I} \widehat{\Delta}_i \geq \sum_{i \in I} y_i,$$

a contradiction. □

By the last lemma and $\mathcal{BR}_0$, we have

$$(1-x_1)\bar{L}_1^0 - C_1 = \cdots = (1-x_n)\bar{L}_n^0 - C_n \geq 0$$

Let $v \equiv (1-x_1)\bar{L}_1^0 - C_1$. Then, $1 - x_i = \frac{v+C_i^0}{\bar{L}_i^0} > 0$. If $v > 0$ then $y_0 = 0$. Because $x_i < 1$, we have $y_i \leq \widehat{\Delta}_i$ for all $i \in [n]$. Thus, we have $y_i = \widehat{\Delta}_i$ for all $i \in [n]$ because otherwise if $y_t < \widehat{\Delta}_t$ for some $t \in [n]$, then $1 = y_0 + y_t + \sum_{i=1, i\neq t}^n y_i < \sum_{i=1}^n \widehat{\Delta}_i = 1$, a contradiction. If, instead, $v = 0$, for all i, we have $x_i = 1 - \eta_i^0 > 0$, which implies $y_i = \widehat{\Delta}_i$. Therefore, $y_0 = 1 - \sum_{i=1}^n y_i = 1 - \sum_{i=1}^n \widehat{\Delta}_i = 0$.

□

Lemma 13. *In every MSNE (x, y) of an IDD game in which $\sum_{i=1}^n \widehat{\Delta}_i > 1$, the probability of no attack $y_0 = 0$.*

Proof. The proof is by contradiction. Suppose $y_0 > 0$. Then, by Claim 6, we have $y_i = \widehat{\Delta}_i$ for all $i \in [n]$, and $1 = \sum_{i=0}^n y_i = \sum_{i=1}^n \widehat{\Delta}_i > 1$, a contradiction. □

Proposition 8. *In every MSNE (x, y) of an IDD game, the probability of no attack $y_0 > 0$ if and only if the game has the property $\sum_{i=1}^n \widehat{\Delta}_i < 1$.*

Proof. The "if" part is Lemma 11. For the "only if" part, the case in which $\sum_{i=1}^n \widehat{\Delta}_i = 1$ follows from Proposition 7; the case in which $\sum_{i=1}^n \widehat{\Delta}_i > 1$ follows from Lemma 13. □

Proposition 9. *In every MSNE (x, y) of an IDD game in which $\sum_{i=1}^n \widehat{\Delta}_i > 1$, no defender is fully investing and some defender is not investing at all (i.e., $x_i = 0$ for some $i \in [n]$).*

Proof. The proof is by contradiction. Proposition 8 yields $y_0 = 0$. Suppose $x_i = 1$ for some $i \in [n]$. Then, by $\mathcal{BR}_i$, $y_i \geq \widehat{\Delta}_i$, and by $\mathcal{BR}_0$ and the fact that $y_0 = 0$, we have $0 > -C_i^0 = M_i(x_i) \geq 0$, which implies $y_i = 0$, a contradiction.

Now suppose $0 < x_i < 1$ for all $i \in [n]$. Then, by $\mathcal{BR}_i$, we have $y_i = \widehat{\Delta}_i$ for all $i \in [n]$. Thus we have $1 = \sum_{i=1}^n y_i = \sum_{i=1}^n \widehat{\Delta}_i > 1$, a contradiction. □

Proposition 10. *The mixed-strategy profile (x, y) is an MSNE of an IDD game in which $\sum_{i=1}^n \widehat{\Delta}_i > 1$ if and only if it satisfies the following properties.*

1. *There is always an attack:* $y_0 = 0$.

2. *There exists a non-singleton, non-empty subset* $I \subset [n]$, *such that* $\min_{i \in I} \bar{M}_i^0 \geq \max_{k \notin I} \bar{M}_k^0$, *if* $I \neq [n]$, *and the following holds.*

   (a) *No defender outside* I *invests or is attacked directly:* $x_k = 0$ *and* $y_k = 0$ *for all* $k \notin I$.

   (b) *Let* $J \equiv \arg\min_{i \in I} \bar{M}_i^0$. *No defender in* J *invests and the probability of that defender being attacked directly is at most the defender's cost-to-expected-loss ratio: for all* $i \in J$, $x_i = 0$ *and* $0 \leq y_i \leq \widehat{\Delta}_i$; *in addition,* $\sum_{i \in J} y_i = 1 - \sum_{t \in I-J} \widehat{\Delta}_i$.

   (c) *Every defender in* $I - J$ *partially invests and has positive probability of being attacked directly equal to the defender's cost-to-expected-loss ratio: for all* $i \in I - J$, $y_i = \widehat{\Delta}_i$ *and*

$$0 < x_i = 1 - \frac{\min_{t \in I} \bar{M}_t^0 + C_i^0}{\bar{L}_i^0} < 1.$$

Proof. For the "if" part, we need to show $(\mathbf{x}, y)$ form mutual best-responses. For all $k \notin I$, $x_k = 0 \in \mathcal{BR}_k(y)$ because $y_k = 0 < \widehat{\Delta}_k$. For all $j \in J$, $x_j = 0 \in \mathcal{BR}_j(y)$ because $y_j \leq \widehat{\Delta}_j$. Finally, for all $i \in I - J$, $x_i \in \mathcal{BR}_i(y_i) = [0, 1]$ because $y_i = \widehat{\Delta}_i$. Hence, we have $x_i \in \mathcal{BR}_i(y_i)$ for all $i \in [n]$. For the attacker, let $v \equiv v(I) \equiv \min_{i \in I} \bar{M}_i^0$. We have for all $k \notin I$, $M_k(x_k) = \bar{M}_k^0 \leq \max_{l \notin I} \bar{M}_l^0 \leq \min_{i \in I} \bar{M}_i^0 = v$, where the first equality holds because $x_k = 0$ and the second inequality by the properties of I. We also have for all $j \in J$, $M_j(x_j) = \bar{M}_j^0 = \min_{i \in I} \bar{M}_i^0 = v$, where the first equality holds because $x_j = 0$ and the second follows from the definition of J. Finally, using simple algebra, we also have for all $i \in I - J$,

$$\begin{aligned} M_i(x_i) &= (1 - x_i)\bar{L}_i^0 - C_i^0 \\ &= \left(\frac{\min_{t \in I} \bar{M}_t^0 + C_i^0}{\bar{L}_i^0} \right) \bar{L}_i^0 - C_i^0 \\ &= \min_{t \in I} \bar{M}_t^0 + C_i^0 - C_i^0 = \min_{t \in I} \bar{M}_t^0 = v. \end{aligned}$$

Hence, we have for all $i \in [n]$, $M_i(x_i) \leq v$. The expected payoff of the attacker under the given mixed-strategy profile is

$$\begin{aligned}\sum_{i=1}^{n} y_i M_i(x_i) &= \sum_{j\in J} y_j M_j(x_j) + \sum_{i\in I-J} y_i M_i(x_i)\\ &= \sum_{j\in J} y_j v + \sum_{i\in I-J} y_i v\\ &= v\left(\sum_{j\in J} y_j + \sum_{i\in I-J} y_i\right)\\ &= v\left(\sum_{i=1}^{n} y_i\right) = v \geq M_i(x_i),\end{aligned}$$

for all $i \in [n]$. Hence, we also have $y \in \mathcal{BR}_0(x)$, and the mixed-strategy profile (x, y) is an MSNE.

We now consider the "only if" part of the proposition. Let (x, y) be an MSNE and let $I \equiv I(y) \equiv \{i \in [n] \mid y_i > 0\}$ be the support of the aggressor's mixed strategy. We now show that I is a non-singleton and non-empty subset of $[n]$.

Claim 7. *$1 < |I| \leq n$.*

Proof. From Proposition 5, we have $I \neq \emptyset$. That I is not a singleton set follows from Lemma 10. □

By Proposition 8, we have $y_0 = 0$. Applying Proposition 9, let $t \in [n]$ be such that $x_t = 0$. Also by Proposition 9, the aggressor achieves a positive expected payoff: $\sum_{i=1}^{n} y_i M_i^0(x_i) = \max_{l=1}^{n} M_l^0(x_l) \geq M_t^0(x_t) = \bar{M}_t^0 > 0$. For all $k \notin I$, because $y_k = 0$, Claim 5 implies $x_k = 0$.

By $\mathcal{BR}_0$, if I is a strict, non-empty and non-singleton subset of $[n]$, we have, for all $i \in I$ and $k \notin I$,

$$\bar{M}_i^0 \geq M_i^0(x_i) = \max_{l\in I} M_l^0(x_l) \geq \bar{M}_k^0 > 0;$$

otherwise, if $I = [n]$, we have, for all $i \in [n]$,

$$M_i^0(x_i) = \max_{l\in[n]} M_l^0(x_l) = M_t^0(x_t) = \bar{M}_t^0 > 0.$$

Let $v \equiv v(I) \equiv \max_{l \in I} M_l^0(x_l)$. Then, the above expressions imply that for all $i \in I$, we have

$$0 < x_i = 1 - \frac{v + C_i^0}{\bar{L}_i^0} < 1.$$

In addition, we have that if I is a strict, non-empty and non-singleton subset of $[n]$, we have,

$$v = \bar{M}_t^0 \geq \min_{i \in I} \bar{M}_i^0 \geq v \geq \max_{k \notin I} \bar{M}_k^0;$$

and if, instead, $I = [n]$, then

$$v = \bar{M}_t^0 = \min_{i \in [n]} \bar{M}_i^0.$$

Hence, we have $v = \min_{i \in I} \bar{M}_i^0$.

Let $J \equiv J(I) \equiv \arg\min_{i \in I} \bar{M}_i^0$. For all $i \in J$, we have $\bar{M}_i^0 = v$, and thus

$$x_i = 1 - \frac{v + C_i^0}{\bar{L}_i^0} = 1 - \frac{\bar{M}_i^0 + C_i^0}{\bar{L}_i^0} = 1 - \frac{\bar{L}_i^0 - C_i^0 + C_i^0}{\bar{L}_i^0} = 0,$$

and by $\mathcal{BR}_i$, we have $0 \leq y_i \leq \widehat{\Delta}_i$.

For all $i \in I - J$, we have $\bar{M}_i^0 > v$, and thus

$$0 = 1 - \frac{\bar{M}_i^0 + C_i^0}{\bar{L}_i^0} < x_i = 1 - \frac{v + C_i^0}{\bar{L}_i^0} < 1,$$

and by $\mathcal{BR}_i$, we have $y_i = \widehat{\Delta}_i$.

Finally, we have $\sum_{i \in J} y_i = 1 - \sum_{i \in I - J} \widehat{\Delta}_i$, because y is a mixed-strategy (i.e, a probability distribution). □

Hence, from the proof of the last proposition we can infer that if the $\bar{M}_l^0$'s form a complete order, then the last condition allows us to search for an MSNE by exploring only $n - 2$ sets, as opposed to 2^{n-2} if done naively.

It turns out a complete order is not necessary. The following claim allows us to safely move all the defenders with the same value of $\bar{M}_i^0$ in a group as a whole inside or outside I.

Claim 8. *Let $I \subset [n]$, such that $I' \subset I$, $|I'| < |I| < n - 1$. Suppose we find an MSNE (x, y) such that $I' = \{i \mid y_i > 0\}$, with the property that $\min_{l \in I'} \bar{M}_l^0 = \max_{k \notin I'} \bar{M}_k^0$. In addition, suppose I satisfies $\min_{l \in I'} \bar{M}_l^0 = \min_{l \in I} \bar{M}_l^0 \geq \max_{k \notin I} \bar{M}_k^0$. Then, we can also find (x, y) using partition I.*

Proof. To simplify the notation, let $v \equiv \min_{l \in I} \bar{M}_l^0 = \min_{l \in I'} \bar{M}_l^0$, $J' \equiv \arg\min_{l \in I'} \bar{M}_l^0$ and $J \equiv \arg\min_{i \in I} \bar{M}_i^0$. The hypothesis implies that (x, y) satisfies the following properties.

$$\begin{aligned}
\text{for all } i \notin I' \text{: } & x_i = y_i = 0 \\
\text{for all } i \in J' \text{: } & x_i = 0 \text{ and } 0 \leq y_i \leq \widehat{\Delta}_i; \\
& \text{also } \sum_{i \in J'} y_i = 1 - \sum_{i \in I' - J'} \widehat{\Delta}_i \\
\text{for all } i \in I' - J' \text{: } & x_i = 1 - \frac{v + C_i^0}{\bar{L}_i^0} \text{ and } y_i = \widehat{\Delta}_i
\end{aligned}$$

We now show that (x, y) also satisfies the constraints when using I with the properties stated in the claim. For that, it needs to satisfy the same expressions as above, but with I' and J' replaced by I and J, respectively.

The first condition holds because $I' \subset I$. The second condition holds for all $i \in J - I'$, because $i \notin I'$ satisfies $x_i = 0$ and $0 \leq y_i = 0 \leq \widehat{\Delta}_i$. It also holds for all $i \in J \cap I'$ because $i \in J$ implies $\bar{M}_i^0 = v$, and because $i \in I'$ and $i \in J'$. For the third condition, note that $I - J \subset I' - J'$ because $i \in I - J$ implies the inequality $\bar{M}_i^0 > v = \max_{k \notin I'} \bar{M}_k^0$; hence, the first inequality in the last expression implies $i \notin J'$, while the equality implies $i \in I'$. □

Proposition 4 follows by combining Propositions 6, 7 and 10. We now discuss properties of the characterization.

4.4.2 Security investment characteristics of MSNE

At equilibrium x^*, if $x_i^* > 0$, the probability of *not* investing is proportional to C_i^0 and *inversely* proportional to $\widehat{p}_i L_i + \sum_{j \in \mathrm{Ch}(i)} \widehat{q}_{ij} L_j$. It is kind of reassuring at equilibrium, which is the (almost-surely) unique stable outcome of the system, the probability of investing increases with the potential loss a player's non-investment decision could cause to the system. Hence, behavior in a stable system implicitly "forces" all players to indirectly account for or take care of their own children. This may sound a bit paradoxical at first given that we are working within a "noncooperative" setting and each player's cost function is only dependent on the investment decision of the player's *parents*. Interestingly, the existence of the attacker in the system is inducing an (almost-surely) unique stable outcome in which an implicit form of "cooperation" occurs. A defenders's best response is independent of their parents, the source of transfer risk, if investment in security does nothing to protect that player from transfers (i.e., $\alpha_i = 1$). This makes sense because the player cannot control the transfer risk. Said differently, there is nothing the player can do to prevent the transfer, even though the original potential for transfers does depend on the parents' investment strategies. In short, rational/optimal noncooperative behavior for each player is not only to protect for the player's own losses but also "cooperate" to protect the player's children.

4.4.3 Relation to network structure

How does the network structure and the equilibrium relate? As seen above, the values of the equilibrium strategy of each player depend on information from the attacker, the player and the player's children. From the discussion in the last paragraph, a player's probability of investing at the equilibrium increases with the expected loss sustained from a "bad event" occurring as a result of a transfer from a player to the player's children.

Let us explore this last point further by considering the case of uniform-transfer probabilities (also studied by Kunreuther and Heal [2003] and Kearns and Ortiz [2004]). In that case, transfer probabilities are only a function of the source, not the destination: $\widehat{q}_{ij} \equiv \widehat{\delta}_i$. The expression for the equilibrium probabilities of those players who have a positive probability of investing would simplify to $x_i^* = 1 - \frac{v+C_i^0}{\widehat{p}_i L_i + \delta_i \sum_{j \in \mathrm{Ch}(i)} L_j}$, for some constant v. The last expression suggests that $\sum_{j \in \mathrm{Ch}(i)} L_j$ differentiates the probability of investing between players. That would suggest that the larger the number of children the larger the probability of investing. A scenario that seems to further lead us to that conclusion is when we make the further assumption of an homogeneous system as first studied in the original IDS paper [Kunreuther and Heal, 2003]: $L_i \equiv L$, $\widehat{p}_i \equiv \widehat{p}$, $\delta_i \equiv \delta$, and $C_i^0 \equiv C^0$ [3] for all players. Then, we would get $x_i^* = 1 - \frac{v+C^0}{L(\widehat{p}+\delta|\mathrm{Ch}(i)|)}$. So the probability of *not* investing, $1-x_i^*$, is inversely proportional to the *number* of children player i has.

On the attacker's equilibrium strategy. The support of the attacker, $I^* \equiv \{i \mid y_i^* > 0\}$, at equilibrium has the following properties: (1) players for which the attacker's cost-to-expected-loss is higher are "selected" first in

[3] Note that this does not mean that the expected loss caused by a player that does not invest but is attacked, $L\,(\widehat{p} + \delta|\mathrm{Ch}(i)|)$, is the same for all players.

the algorithm; (2) if the size of that set is t, and there is a lower bound on $\widehat{\Delta}_i > \widehat{\Delta}$, and $\sum_{i=1}^{n} \widehat{\Delta}_i > 1$, then $t < 1/\widehat{\Delta}$ is an upper-bound on the number of players that could potentially be attacked; (3) if we have a game with homogeneous parameters, then the probability of an attack will be uniform over that set I^*; and (4) all but one of the players in that set I^* invest in security with some non-zero probability (almost surely).

Algorithm 4: Compute all MSNE of a single-attack transfer-vulnerable IDD game.

```
Input  : A SATV IDD game $\mathcal{G} = (G = ([n], E), \widehat{\mathbf{p}}, \widehat{\mathbf{Q}}, \mathbf{L}, \mathbf{C}, \mathbf{C}^0)$
Output: he set $\mathcal{NE}$ of all MSNE of $\mathcal{G}$
foreach $i = 1$ to $n$ do
    $\widehat{\Delta}_i \leftarrow \frac{C_i}{\widehat{p}_i L_i}$
    $\mathrm{Ch}(i) \leftarrow \{j \in [n] \mid (i,j) \in E\}$
    $\bar{L}_i^0 \leftarrow \widehat{p}_i L_i + \sum_{j \in \mathrm{Ch}(i)} \widehat{q}_{ij} L_j$
    $\eta_i^0 \leftarrow C_i^0 / \bar{L}_i^0$
    $\bar{M}_i^0 \leftarrow \bar{L}_i^0 - C_i^0$
end foreach
if $\sum_{i=1}^{n} \widehat{\Delta}_i < 1$ then
    Assign to $\mathcal{NE}$ the output of call to subroutine for this case given in
    Algorithm 5 with input $n$, $\boldsymbol{\eta}^0$, $\widehat{\boldsymbol{\Delta}}$
end if
if $\sum_{i=1}^{n} \widehat{\Delta}_i = 1$ then
    Assign to $\mathcal{NE}$ the output of call to subroutine for this case given in
    Algorithm 6 with input $n$, $\widehat{\boldsymbol{\Delta}}$, $\bar{\mathbf{L}}^0$, $\mathbf{C}^0$
end if
if $\sum_{i=1}^{n} \widehat{\Delta}_i > 1$ then
    Assign to $\mathcal{NE}$ the output of call to subroutine for this case given in
    Algorithm 7 with input $n$, $\widehat{\boldsymbol{\Delta}}$, $\bar{\mathbf{L}}^0$, $\mathbf{C}^0$, $\bar{\mathbf{M}}^0$
end if
return $\mathcal{NE}$
```

Algorithm 5: Subroutine to compute the unique MSNE of a single-attack transfer-vulnerable IDD game with $\sum_{i=1}^{n} \widehat{\Delta}_i < 1$.

```
Input  : $n$, $\widehat{\boldsymbol{\Delta}}$, $\boldsymbol{\eta}^0$
Output: The unique MSNE for this case as the set $\mathcal{NE}$
$S \leftarrow 0$
foreach $i = 1$ to $n$ do
    $x_i \leftarrow 1 - \eta_i^0$
    $y_i \leftarrow \widehat{\Delta}_i$
    $S \leftarrow S + y_i$
end foreach
$y_0 \leftarrow 1 - S$
$\mathcal{NE} \leftarrow \{(x, y)\}$
return $\mathcal{NE}$
```

Algorithm 6: Subroutine to compute (a simple linear representation of) all MSNE of a single-attack transfer-vulnerable IDD game with $\sum_{i=1}^{n} \widehat{\Delta}_i = 1$.

```
Input  : $n$, $\widehat{\boldsymbol{\Delta}}$, $\bar{\mathbf{L}}^0$, $\mathbf{C}^0$
Output: The set $\mathcal{NE}$ of all MSNE for this case
foreach $i = 1$ to $n$ do
    $y_i \leftarrow \widehat{\Delta}_i$
end foreach
$y_0 \leftarrow 0$
$\mathcal{X} \leftarrow \{x \geq 0 \mid (1 - x_1)\bar{L}_1^0 - C_1^0 = \cdots = (1 - x_n)\bar{L}_n^0 - C_n^0 \geq 0\}$
$\mathcal{NE} \leftarrow \mathcal{X} \times \{y\}$
return $\mathcal{NE}$
```

4.4.4 Computing All MSNE Efficiently

We now describe an algorithm to compute *all* MSNE in single-simultaneous-attack transfer-vulnerable IDD games that falls off Proposition 4. We begin by noting that the equilibrium in the case of IDD games with $\sum_{i=1}^{n} \widehat{\Delta}_i \leq 1$, corresponding to cases 1 and 2 of the proposition, has essentially an analytic closed-form. Hence, we concentrate on the remaining and most realistic case in large-population games of $\sum_{i=1}^{n} \widehat{\Delta}_i > 1$. We start by sorting the indices of

Algorithm 7: Subroutine to compute (a simple simplex representation of) all MSNE of a single-attack transfer-vulnerable IDD game with $\sum_{i=1}^{n} \widehat{\Delta}_i > 1$.

```
Input  : $n$, $\widehat{\mathbf{\Delta}}$, $\bar{\mathbf{L}}^0$, $\mathbf{C}^0$, $\bar{\mathbf{M}}^0$
Output: The set $\mathcal{NE}$ of all MSNE for this case
$(\mathbf{Val}, \mathbf{Idx}) \leftarrow \mathrm{sort}(\bar{\mathbf{M}}^0, \text{'descending'})$
$t \leftarrow 0$
$S \leftarrow 0$
while $t = n$ or $S \geq 1$ do
    $t \leftarrow t + 1$
    $S \leftarrow S + \widehat{\Delta}_{\mathrm{Idx}(t)}$
end while
$k \leftarrow t$
$S \leftarrow S - \widehat{\Delta}_{\mathrm{Idx}(t)}$
$v \leftarrow \mathrm{Val}(t)$ while $\mathrm{Val}(t) = v$ and $t < n$ do
    $t \leftarrow t + 1$
end while
foreach $i = 1$ to $k - 1$ do
    $l \leftarrow \mathrm{Idx}(i)$
    $x_l \leftarrow 1 - \frac{v + C_l^0}{\bar{L}_l^0}$
    $y_l \leftarrow \widehat{\Delta}_l$
end foreach
$O \leftarrow \emptyset$
foreach $i = k$ to $t - 1$ do
    $l \leftarrow \mathrm{Idx}(i)$
    $x_l \leftarrow 0$
    $O \leftarrow O \cup \{l\}$
end foreach
foreach $i = t$ to $n$ do
    $l \leftarrow \mathrm{Idx}(i)$
    $x_l \leftarrow 0$
    $y_l \leftarrow 0$
end foreach
$\mathcal{Y}_O \leftarrow \{y_O \mid 0 \leq y_i \leq \widehat{\Delta}_i,$ for all $i \in O$, and $\sum_{i \in O} y_i = 1 - S\}$
$\mathcal{NE} \leftarrow \{x\} \times \{y_{-O}\} \times \mathcal{Y}_O$
return $\mathcal{NE}$
```

the internal players in descending order based on the $\bar{M}_i^0$'s. Let $\text{Val}(l)$ and $\text{Idx}(l)$ be the lth value and index in the resulting sorted list, respectively. Find t such that $1 - \widehat{\Delta}_{\text{Idx}(t)} \leq \sum_{l=1}^{t-1} \widehat{\Delta}_{\text{Idx}(l)} < 1$. Let $k = \arg\max\{l \geq t \mid \text{Val}(l) = \text{Val}(t)\}$ (i.e., continue down the sorted list of values until a change occurs). For $i = 1, \ldots, t-1$, let $l = \text{Idx}(i)$ and set $x_l^* = 1 - \frac{\text{Val}(t)+C_l^0}{\bar{L}_l^0}$ and $y_l^* = \widehat{\Delta}_l$. For $i = k+1, \ldots, n$, let $l = \text{Idx}(i)$ and set $x_l^* = 0$ and $y_l^* = 0$. For $i = t, \ldots, k$, let $l = \text{Idx}(i)$ and set $x_l^* = 0$. Finally, represent the simplex defined by the following constraints: for $i = t, \ldots, k$, let $l = \text{Idx}(i)$ and $0 \leq y_l^* \leq \widehat{\Delta}_l$; $\sum_{i=t}^{k} y_{\text{Idx}(i)}^* = 1 - \sum_{i=1}^{t-1} \widehat{\Delta}_{\text{Idx}(i)}$. The running time of the algorithm is $O(n \log n)$ (because of sorting).

Theorem 12. *There exists a polynomial-time algorithm to compute all MSNE of a single-simultaneous-attack transfer-vulnerable IDD game.*

In cases in which the equilibria is not unique, it can be generated via simple sampling of either a simple linear system or a simplex. In either case, one can compute a single MSNE from that infinite set in polynomial time.

Let us revisit the types of games that may have an infinite MSNE set. Note that the case in which $\sum_{i=1}^{n} \widehat{\Delta}_i = 1$ has (Borel) measure zero and is quite brittle (i.e., adding or removing a player breaks the equality). For the case in which $\sum_{i=1}^{n} \widehat{\Delta}_i > 1$, if the value of the $\bar{M}_i^0$'s are distinct, [4] then there is a unique MSNE! Algorithm 4 provides pseudocode of the algorithm resulted from the characterization.

[4] Distinct $\bar{M}_i^0$'s for the set of defenders at which the sum goes over one is sufficient to guarantee unique MSNE.

4.5 Experiments: Computing an ϵ-MSNE using BRGD

In the previous section, we established theoretical characteristics and computational tractability of single-simultaneous-attack IDD games with the highest transfer vulnerability parameter: $\alpha_i = 1$. In this section, partly motivated by security problems in cyberspace, we concentrate instead on empirically evaluating the other extreme of transfer vulnerability: games with low α_i values (i.e., near 0), so that investing in security considerably reduces the transfer risk.

Our main objectives for the experiments presented here are (1) to demonstrate that a simple heuristic, *best-response-gradient dynamics (BRGD)*, is practically effective in computing an (approximate) MSNE in a very large class of IDD games with realistic Internet-scale network graphs in a reasonable amount of time for cases in which the transfer vulnerabilities α_i's are low; and (2) to explore the general structural and computational characteristics of (approximate) MSNE in such IDD games, including their dependence on the underlying network structure of the game (and approximation quality).

BRGD is a well-known technique from the work on learning in games [Fudenberg and Levine, 1998, Singh et al., 2000, Kearns and Ortiz, 2004, Heal and Kunreuther, 2005a, Kearns, 2005]. Here, we use BRGD as a tool to *compute* an ϵ-approximate MSNE, which is a mixed-strategy profile with the property that the gain in utility (or reduction in cost) of any individual from unilaterally deviating from their prescribed mixed-strategy is no larger than ϵ. In particular, a mixed-strategy profile (x, y) is an ϵ-MSNE if for all sites i, $M(x, y) \leq M(0, y) + \epsilon$ and $M(x, y) \leq M(1, y) + \epsilon$ and $U(x, y) \geq U(x, y') - \epsilon$ for $1 \geq \epsilon \geq 0$ for all $y' \in [0, 1]^n$. A 0-approximate MSNE is an exact MSNE. Notice that in the discussion of ϵ-MSNE, the cost functions and utility are required to normalized between zero and one.

We obtained the latest version (March 2010 at the time) of the real structure and topology of the Autonomous Systems (AS) in the Internet from DIMES (netdimes.org) [Shavitt and Shir, 2005]. The AS-level network has $27,106$ nodes (683 isolated) and $100,402$ directed edges; the graph length (diameter) is $6,253$, the density (number of edges divided by number of possible edges) is 1.9920×10^{-5}, and the average (in and out) degree is 3.70, with $\approx 76.93\%$ and 2.59% of the nodes having zero indegree and outdegree, respectively. Figure 4.1 shows the indegree and outdegree distribution and

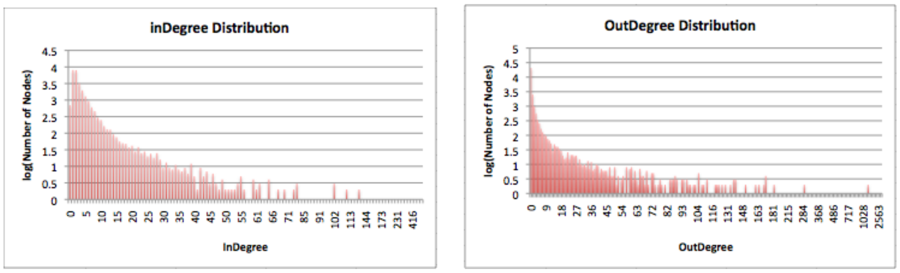


Figure 4.1: **Histograms of Indegree and Outdegree of the Nodes of the Autonomous Systems from DIMES.** The bar graphs show (the logarithm (base 10) of) the number nodes with a particular outdegree (left) and indegree (right) value. (The graphs only show the in/out degrees with a non-zero number of nodes.)

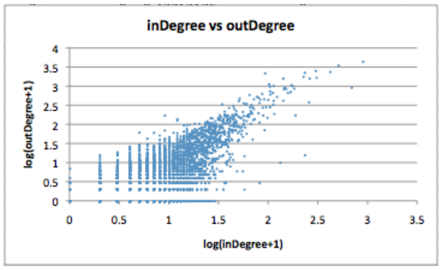


Figure 4.2: **Indegree and Outdegree of the Nodes of the Autonomous Systems from DIMES.** The scatter plot shows the indegree and outdegree pairs of the AS nodes in logarithmic (base 10) scale.

Figure 4.2 shows the scatter plot of indegree and outdegree of the graph. *All the IDD games in the experiments presented in this section have this network structure.*

For simplicity, we call *Internet games* the class of IDD games with the AS-level network graph structure. We considered various settings for model parameters of Internet games. In particular, we generate the values of the parameters according to Table 4.1. From the initialization, we have two instances of IDD games. The first instance of IDD games has its parameters' values generated uniformly at random within some ranges. The second instance of IDD games has its parameters' values generated from the expec-

Model Parameters	Fixed: U = 0.5 Random: U $\sim$ Uniform([0,1])
α_i	$U/20$
L_i	$10^8 + (10^9) * U$
C_i	$10^5 + (10^6) * U$
$\widehat{p}_i$	$0.9 * \frac{\widetilde{p}_i}{\widetilde{p}_i + \sum_{k \in \mathrm{Ch}(i)} \widetilde{q}_{ik}}$
$\widehat{q}_{ij}$	$0.9 * \frac{\widetilde{q}_{ij}}{\widetilde{p}_i + \sum_{k \in \mathrm{Ch}(i)} \widetilde{q}_{ik}}$
z_i	$0.2 + U/5$
$\widetilde{p}_i$	$0.8 + U/10$
$\widetilde{q}_{ij}$	$z_i \frac{\lvert\mathrm{Ch}(j)\rvert + \lvert\mathrm{Pa}(j)\rvert}{\sum_{k \in \mathrm{Ch}(i)} \lvert\mathrm{Ch}(k)\rvert + \lvert\mathrm{Pa}(k)\rvert}$
C_i^0	10^6

Table 4.1: **Internet Games' Model Parameters.**

tation of the random instance. Therefore, the second instance of IDD games is fixed and has only one single instance.

The attacker's cost-to-attack parameter for each node i is always held constant: $C_i^0 = 10^6$. For each run of each experiment, we ran BRGD with randomly-generated initial conditions (i.e., random initializations of the players' mixed strategies): $x_i \sim \text{Uniform}([0,1])$, i.i.d. for all i, and y is a probability distribution generated uniformly at random, and independent of $\mathbf{x}$, from the set of all probability mass functions over $n+1$ events. [5] The initialization of the transfer-probability parameters of a node essentially gives higher transfer probability to children with high (total) degree (because they are potentially "more popular"). The initialization also enforces $\widehat{p}_i + \sum_{j \in \mathrm{Pa}(i)} \widehat{q}_{ji} = 0.9$. Other initializations are possible but we did not explore then here.

4.5.1 Computing an ϵ-MSNE using BRGD

Given the lack of theoretical guarantees on the convergence rate of BRGD, we began our empirical study by evaluating the convergence and computation/running-time behavior of BRGD on Internet games. We ran ten simulations for each ϵ value and recorded the number of iterations until convergence (up to $2,000$

[5] Recall the probability of no attack $y_0 = 1 - \sum_{i=1}^{n} y_i$.

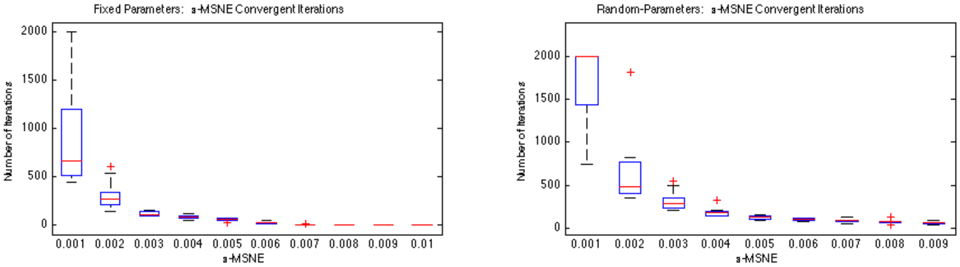


Figure 4.3: **Convergence Rate of Learning Dynamics.** The plots above present the number of iterations of BRGD as a function of ϵ under the two experimental conditions: Internet games with fixed (top) and randomly-generated parameters (bottom). Applying MSE regression to the top and bottom graphs, we obtain a functional expression for the number of iterations $N^F(\epsilon) = 0.00003\epsilon^{-2.547}$ ($R^2 = 0.90415$) and $N^R(\epsilon) = 0.0291\epsilon^{-1.589}$ ($R^2 = 0.9395$), respectively (i.e., low-degree polynomials of $1/\epsilon$).

iterations). Figure 4.3 presents the number of iterations taken by BRGD to compute an ϵ-MSNE as a function of ϵ. All simulations in this experiment converged (except for $\epsilon = 0.001$: 2 and all of the runs for single and randomly-generated instances, respectively, did not). Each iteration took roughly 1-2 sec. (wall clock). Hence, we can use BRGD to consistently compute an ϵ-MSNE of a 27K-players Internet game in a few seconds!

We now concentrate on the empirical study of the *structural* characteristics of the ϵ-MSNE found by BRGD. We experimented on both the single and randomly-generated Internet game instances. We discuss the typical behavior of the attacker and the sites in an ϵ-MSNE, and the typical relationship between ϵ-MSNE and network structure.

A Single Internet Game

We first studied the characteristics of the ϵ-MSNE of a single Internet game instance. The only source of randomness in these experiments comes from BRGD's initial conditions (i.e., the initialization of the mixed strategies $\mathbf{x}$ and $\mathbf{y}$). BRGD consistently found *exact* MSNE (i.e., $\epsilon = 0$) in *all* runs.

Players' equilibrium behavior. In fact, we consistently found that the attacker always displays only two types of "extreme" equilibrium behavior,

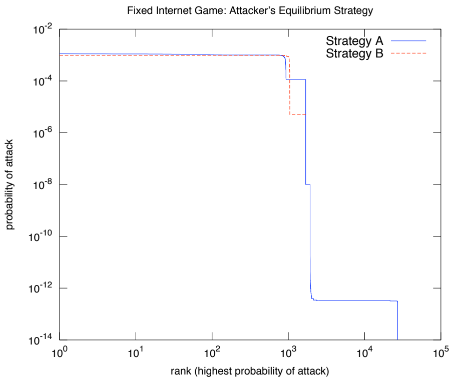


Figure 4.4: **Attacker's Equilibrium Strategy on an Internet Game Instance (Fixed).** The graph shows the values of $y_i^* > 0$ for each node i, sorted in decreasing order (in log-log scale), for attacker's Strategy A (blue/denser-dots line) and Strategy B (red/sparser-dots line) at an MSNE of the single instance of the Internet game.

corresponding to the two kinds of MSNE BRGD found for the single Internet game: place positive probability of a direct attack to either *almost all* nodes (Strategy A) or a *small subset* (Strategy B). Figure 4.4 shows a plot of the typical probability of direct attack for those two equilibrium strategies for the attacker when BRGD stops. In both strategies, a relatively small number of nodes (about 1K out of 27K) have a reasonably *high* (and near *uniform*) probability of direct attack. In Strategy A, however, *every* node has a positive probability of being the target of a direct attack, albeit relatively very low for most; this is contrast to Strategy B where *most* nodes are fully immune from a direct attack. Interestingly, *none* of the nodes invest in either MSNE: $x_i^* = 0$ for all nodes i. Thus, in this particular Internet game instance, *all* site nodes are willing to risk an attack!

Relation to network structure. We found that the nodes with (relatively) high probability of direct attack are at the "fringe" of the graph (i.e., have low or no degree). In Strategy A, fringe nodes (with mostly 0 or 1 outdegree) have relatively higher probability of direct attack than nodes with higher outdegree. Similarly, in Strategy B, the small subset of nodes that are potential target of a direct attack have relatively low outdegree (mostly 0,

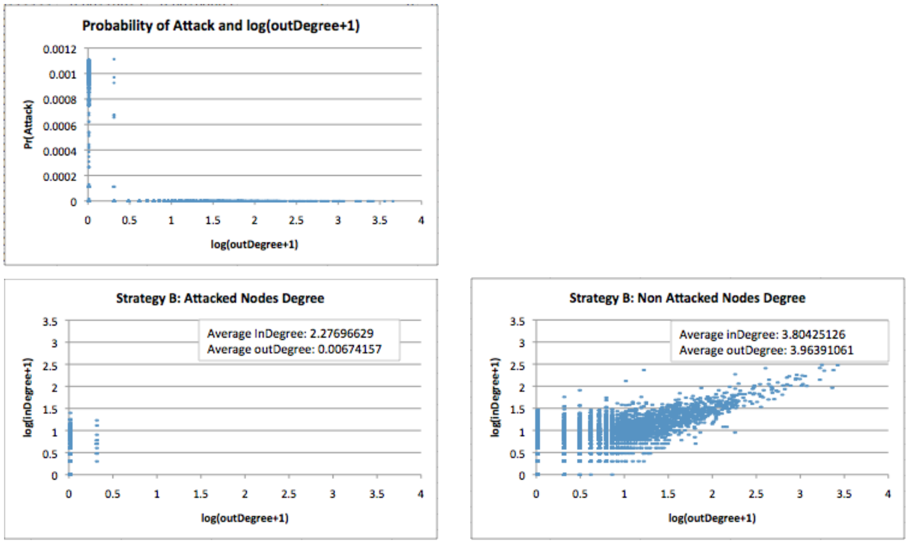


Figure 4.5: **Attacker's Equilibrium Strategy and the Degrees of the Nodes.** The top graph, which depicts Strategy A (all 27106 nodes), shows the probability of attack (y-axis) of a node and its corresponding outdegree (x-axis) in logarithmic (base 10) scale. The below graphs show the indegree (y-axis) of a node and its corresponding outdegree (x-axis) in logarithmic (base 10) scale of Strategy B: the graphs on the left and right consist of the (1780) nodes with nonzero probability of attack and the (25326) nodes with zero probability of attack, respectively.

and 0.0067 on average; this is in contrast to the average outdegree of 3.9639 for the nodes immune from direct attack). Figure 4.5 shows the relation between the probability of attack and outdegree and the relation between the indegree and outdegree of a typical simulation runs for strategy A and for Strategy B as described above, respectively. We emphasize that these observations are consistent throughout all runs of the experiment. In short, we consistently found that the nodes with low outdegree are more likely to get attacked directly in the single Internet game instance.

Randomly-Generated Internet Games

We now present results from experiments on randomly-generated instances of ten Internet games, a single instance for each $\epsilon \in \{0.001, 0.002, \ldots, 0.009\}$.

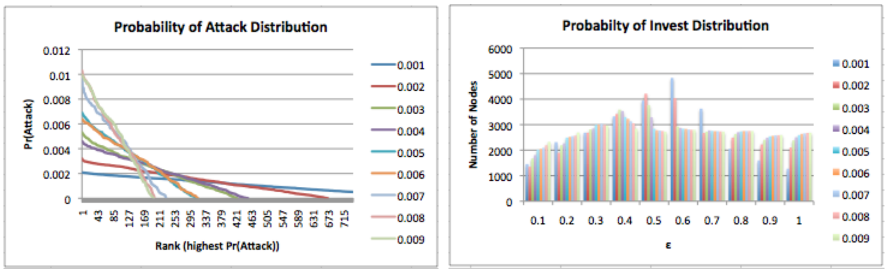


Figure 4.6: **Attacker's and Site's ϵ-MSNE Strategies for a Randomly-Generated Internet Game.** The graphs show the empirical distributions of the probability of attack (top) and histograms of the probability of investment (bottom), for different ϵ-value conditions encoded in the right-hand side of the plots (i.e., from 0.001 to 0.009). In both graphs, the distributions and histograms found for each ϵ value considered are drawn in the same corresponding graph superimposed. The top graph plots the distribution of y_i where the nodes are ordered decreasingly based on the y_i value. The bottom bar graph shows histograms of the probability of investing in defense/security measures based on the following sequence of 10 ranges partitioning the unit interval: $([0, 0.1], (0.1, 0.2], ..., (0.9, 1])$.

For simplicity, we present the result of a single BRGD run on each instance. [6]

Behavior of the players. Figure 4.6 shows plots of the attacker's probability of direct attack and histograms of the nodes's probability of investment in a typical run of BRDG on a randomly-generated Internet game instance for each ϵ value.

The plots suggest that approximate MSNE found by BRGD is quite sensitive to the ϵ value: as ϵ decreases, the attacker tends to "spread the risk" by selecting a larger set of nodes as potential targets for a direct attack,

[6] While some results presented here are for a single instance of the Internet game for each ϵ, the results are typical of multiple instances. Our observations are robust to the experimental randomness in both the Internet game parameters and the initialization of BRGD. For the sake of simplicity of presentation, we discuss results based on a single instance of the Internet game, and in some cases based on a single BRGD run. Note that, for each ϵ value we considered, the Internet game parameters remain constant within different BRGD runs. BRGD always converged within $2,000$ iterations (except 6 runs for $\epsilon = 0.001$).

thus lowering the probability of a direct attack on any individual node; the nodes, on the other hand, tend to deviate from (almost) fully investing and (almost) not investing to a more uniform mixed strategy (i.e., investing or not investing with roughly equal probability).

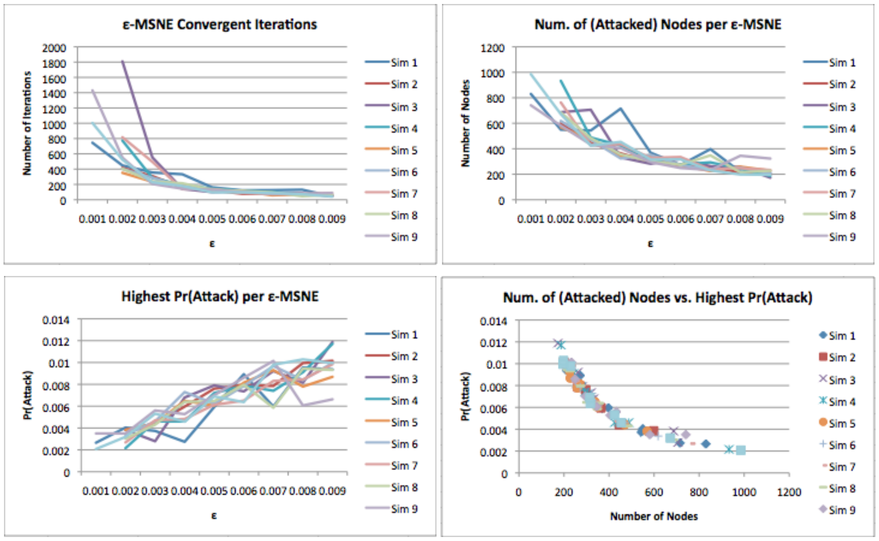


Figure 4.7: **Attacker's Strategy at ϵ-MSNE.** The x-axis of the top left, top right, and bottom left represents the ϵ value and their y-axis represents the number of iterations until convergence (or 2000 iterations max) to some ϵ-MSNE, the number of nodes that are being attacked, and the highest probability of attack, respectively. The bottom right scatter plot shows the relation between the number of nodes that are being attacked and the highest probability of attack in x-axis and y-axis, respectively.

A more thorough study confirms the above observation of the attacker and it is illustrated by Figure 4.7. Figure 4.7 shows: (a) the number of iterations taken by the BRGD for ϵ-MSNE to converge (top left); (b) the number of nodes that are being targeted (top right); (c) the highest probability of attack (bottom left); and (d) the scatter plot of the nodes that are being targeted and the highest probability of attack (bottom right) for each of the ten simulations. From this figure, we observe that, as ϵ decreases, (1) the number of iterations takes for an ϵ-MSNE to converge increases (top left), (2) the number of nodes that are being targeted increases (top right), and (3) the highest probability of attack decreases (bottom left). From the bottom right

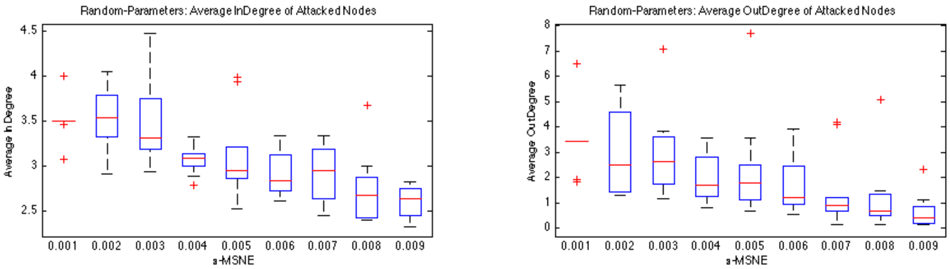


Figure 4.8: **Attacker's ϵ-MSNE Strategy vs. Node Degrees.** Average indegree (top) and outdegree (bottom) of nodes potentially attacked in terms of the ϵ-MSNE.

graph of Figure 4.7, we observe that there is a negative correlation between the number of nodes that are being targeted and the highest probability of an attack: as the highest probability of an attack increases, the number of nodes that are being targeted decreases.

A possible reason to explain the behavior of the sites is that as more nodes become potential targets of a direct attack, more nodes with initial mixed strategies close to the "extreme" (i.e., very high or very low probabilities of investing) will move closer to a more uniform (and thus less "predictable") investment distribution.

Relation to network structure. Figure 4.8 presents some experimental results on the relationship between network structure and the attacker's equilibrium behavior. The graphs show, for each ϵ value, the average indegree and outdegree, across the BRGD runs of the ten randomly-generated instances Internet games, of those nodes that are potential targets of a direct attack at an ϵ-MSNE. In general, both the average indegree and outdegree of the nodes that are potential targets of a direct attack tend to increase as ϵ decreases. One possible reason for this finding could be the fact that the values of α_i generated for each player are relatively low (i.e., uniformly distributed over $\left[0, \frac{1}{40}\right]$); yet, interestingly, such behavior and pattern, is exact opposite of the theory for the case $\alpha_i = 1$.

4.5.2 Case Study: A randomly-generated instances of an Internet games at 0.005-MSNE

In this subsection, we provide a detail topological study of a randomly-generated instances of an Internet games at 0.005-MSNE.

Topological structure of an attack to the Internet. In Figure 4.9, we plot the topological structure of the top sites (in this case 402) with the highest y_i and their immediate neighbors at 0.005-MSNE. Notice that there are a few isolated nodes and a few small ”node-parent-children” networks, but in general, the largest network component tends to have a cluster-like structure. Figure 4.9 also shows the number of connected components of the network for the subgraph of the nodes that are most likely to be attacked (and their neighbors) as well as the probability of investing of all nodes in the network, along with some additional properties of the graphs.

Figure 4.10 and Figure 4.11 show the indegree and outdegree of the (402) non-zero y_i nodes and the remaining (26704) zero y_i nodes, respectively. We did not observe in our experiments any strong relationship between the y_i's or x_i's in the ϵ-MSNE we found and the corresponding indegree or outdegree of the node i. However, we observed that, among the nodes with non-zero probability of attack, there was a slight tendency for those nodes with the lowest probability of attack to also have low outdegree and for those nodes with the highest probability of investing to also have low outdegree, but that tendency did not seem significant enough.

As mentioned earlier, the behavior of the players are quite sensitive to the ϵ value. Therefore, this could be one of the reasons that these nodes (with the highest y_i) have low outdegree.

4.6 Conclusion and Open Problem

In this chapter, we have introduced IDD games, a new class of security games in which the attacker is explicitly model. Under various assumptions, we show that there is no PSNE in the IDD games and we can compute all MSNE of the IDD games. Despite the lack of algorithm on the general IDD games, we use a well-known heuristic to show the behaviors of the sites and the attacker at approximate NE. Below, we present a few possible extensions and modifications of our model and some open problems.

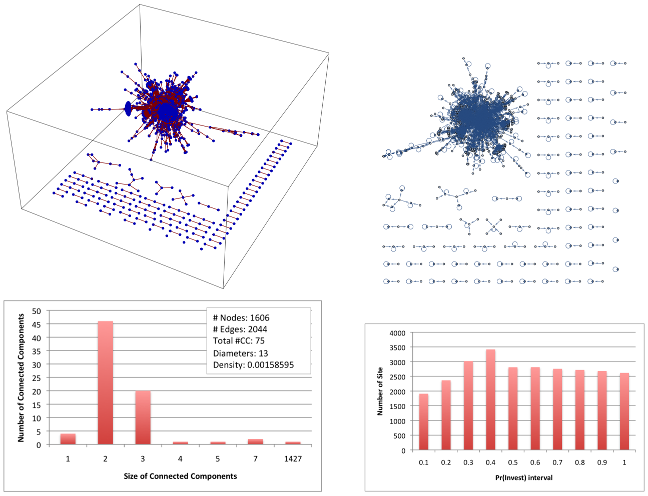


Figure 4.9: **The Structure of an Attack to the Internet.** The 3-d graph top left corresponds to the top 402 Internet AS level nodes most likely to be attacked according to our model at 0.005-MSNE, and their neighbors (i.e., both parent and children family). The graph on the top right is a 2-d projection of the 3-d graph on the top left. The self-loops mark the nodes that are actually attacked. For the most part, the graph structures exhibit very dense clustering. The bar graph on the bottom left corresponds to the number of connected components (CC) of the top 402 Internet AS level nodes that are most likely to be attacked. The bar graph on the bottom right shows the number of nodes with the probability of investing in defense/security measures within the range of $([0, 0.1], (0.1, 0.2], ..., (0.9, 1])$. Some properties of the graph corresponding to the network structure are shown on the upper corner of the bottom left graph The graph consists of 1606 nodes, 2044 edges, and 75 CC. Out of the 75 CC, the largest CC contains 1427 nodes and the smallest CC consists of just one node (there are only four of them). There are 46 of 2-CC (CC with only 2 nodes), 20 of 3-CC, 1 of 4-CC, 1 of 5-CC, and 2 of 7-CC. The diameters and density of the graphs are 13 and 0.002, respectively.

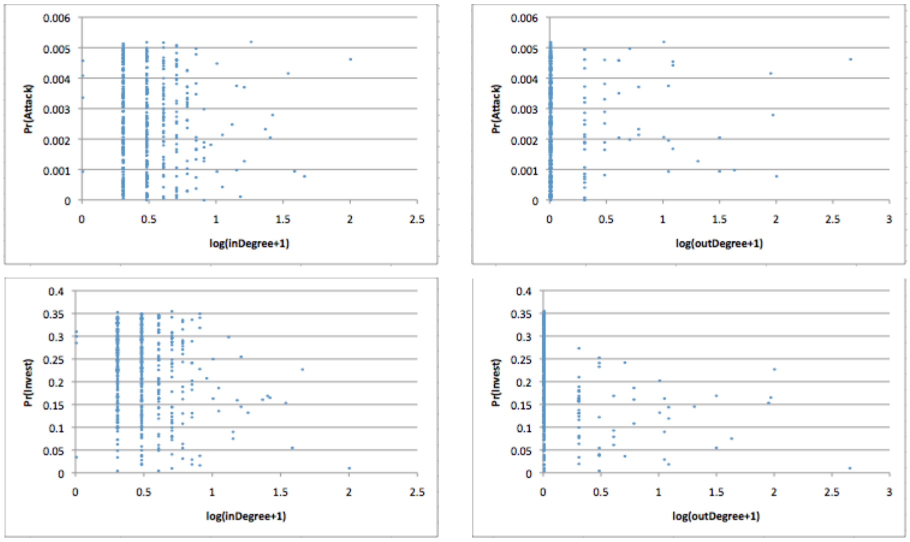


Figure 4.10: **Attacker's Equilibrium Strategy vs Degree of the Nodes at 0.05-MSNE** These are plots on the 402 nodes with the highest y_i. The two graphs on top show the corresponding y_i (y-axis) and its indegree and outdegree in logarithmic (base 10) scale. Similarly, the two graphs at the bottom show the corresponding x_i (y-axis) and its indegree and outdegree in logarithmic (base 10) scale.

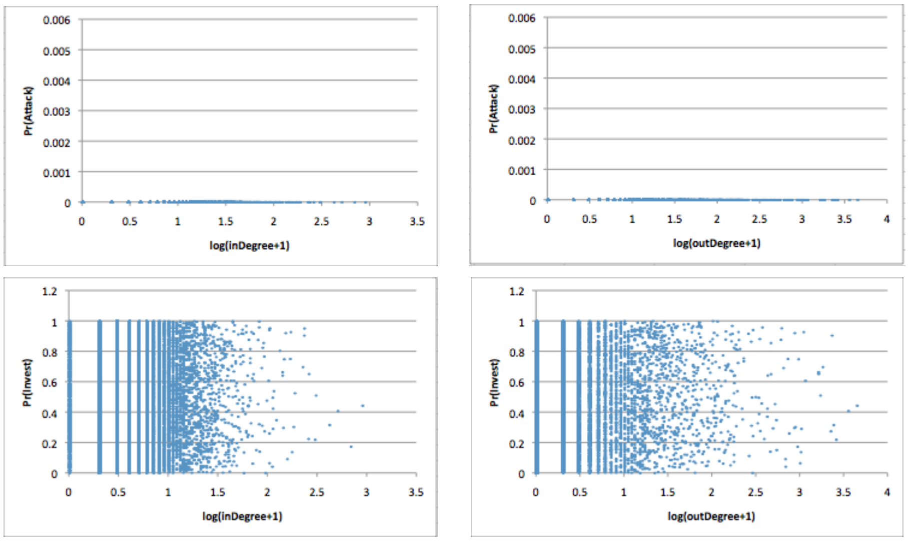


Figure 4.11: **The degrees and strategies of the non-targeted nodes.** These are plots on the remaining 26704 nodes with zero y_i. The two graphs on top show the corresponding y_i (y-axis) and its indegree and outdegree in logarithmic (base 10) scale. Similarly, the two graphs at the bottom show the corresponding x_i (y-axis) and its indegree and outdegree in logarithmic (base 10) scale.

Attackers Can Affect Transfer Probabilities. We could extend the strategy space of the attacker by allowing the attacker to affect transfer. One particular instantiation of this idea is to have the network graph *edges* represent the attacker's targets, as opposed to just the node. The attacker's pure strategies would now be based on the edges (i, j), such that binary action variable b_{ij} would now represent the attack, taking a value of one if the attackers wants to attack j but only via a transfer from i.

Multiple Attackers with Multiple Attacks. While dealing with multiple attackers is outside the scope of this paper, we have in fact extended the model in a natural way in that direction. However, we were able to extend the representation results, but not the characterization or computational/algorithmic results. We leave that endeavor for future work. In principle, the best-response gradient dynamics can also be used as a heuristic in the multiple attackers' case.

Open Problems. A thorough characterization of the equilibria of interdependent defense games is lacking, specially for the case of multiple potential attacks by multiple aggressors. Also, we need a better understanding of the effect of network structure of the game and restrictions on the aggressors' available strategies on the equilibria of the game.

Many computational problems in the context of interdependent defense games remain open.

1. What is the computational complexity of the problem of computing equilibria of interdependent defense games with arbitrary transfer vulnerability? (e.g., a single, multiple or all MSNE? MSNE with particular properties?)

2. What is the computational complexity of the problem of identifying "influential" agents, in the sense of Irfan and Ortiz [2014] (see also, Kleinberg [2007] and the references therein)?

3. How is the complexity affected by network structure or restrictions on the aggressors' available strategies? For example, what if the network graph is some type of chain, cycle or tree?

Chapter 5

Computing Approximate Nash Equilibrium In Interdependent Defense Games[1]

Given that there is no PSNE in any IDD games, we shift our focus to computing an MSNE. In the previous chapter, we provide an algorithm to compute all MSNE in an instance of IDD games where $\alpha_i = 1$ for all sites i. The interpretation is that investment cannot protect the sites from indirect risk. However, there is no result for the harder case of general α_i.

5.1 On the Complexity of Computing an MSNE

Here, we consider the computational complexity of computing an MSNE in general α-IDD games. A closer look at the model reveals something interesting about IDD games: we can view computing an MSNE in IDD games as a two-part process. Given an attacker's strategy, we need to determine the MSNE of the underlying game of the sites, or *sites-game* for short. The sites-game could have many MSNE and each MSNE could yield a different utility for the attacker (and the sites). Naively, the attacker can verify whether each of the MSNE is in the attacker's best response. Clearly, doing so depends on whether we can efficiently compute all MSNE in the sites-game, which of course depends on the given attacker's strategy. For example, if $\sum_{i=1}^{n} y_i = 0$,

[1] A part of this chapter has appeared in the proceedings of the *Twenty-ninth AAAI Conference on Artificial Intelligence (AAAI 2015).*

then the sites-game would have 'none invest' as the only outcome, because of Assumption 2 in Chapter 4.2. In the below, we will continue to study the computational question of computing (approximate) MSNE in IDD games under the model assumptions as specified and discussed in Chapter 4.2.

Our goal here is to show that there is an instance of IDD games, and an attacker's strategy in that instance, such that should we fix that attacker's strategy, we cannot compute all of the MSNE efficiently in the underlying sites-game, unless $P = NP$. The implication is that the existence of an efficient algorithm to compute an MSNE of IDD games based on the iterative process just described, of checking whether each attacker's strategy can be part of an MSNE, would be unlikely.

To formally prove that we cannot always compute all of the MSNE in an instance of the sites-games, as induced by an IDD game and an attacker's strategy, efficiently, we consider the *Pure-Nash-Extension problem* [Kearns and Ortiz, 2004] for binary-action n-player games, which is NP-complete. Recall that the problem takes a description of the game and a *partial* assignment $a \in \{0, 1, *\}^n$ as input. We want to determine whether there is a *complete* assignment $b \in \{0, 1, *\}^n$ consistent with a. Note that proving that computing an MSNE in IDD games is PPAD-complete would be more appropriate, since there is always an MSNE, but we will leave this for future work.

Theorem 13. *Consider a variant of IDD games in which $\sum_{i=1}^{n} R_i/\hat{p}_i \leq 1$. There is an attacker's strategy y such that if we fix y, then the Pure-Nash Extension problem for the induced n-player sites-game is NP-complete.*

Proof. First, we construct a graph structure and set the values of the parameters to define the IDD game based on an NP-complete problem. Next, we show that if y exists, then the induced sites-game solves the NP-complete problem. Finally, we show that such a y exists.

We first define the notations that will be used in the proof. In particular, we consider the problem of determining whether there is a PSNE in the sites-games while fixing some strategies of some players. More specifically, we denote the instances with PSNE as

$$\begin{aligned}\text{sites-game} \;=\; & \{\; ([n], (C_i)_{i\in[n]}, (\alpha_i)_{i\in[n]}, (L_i)_{i\in[n]}, (\widehat{p}_i)_{i\in[n]}, (\widehat{q}_{ji})_{j,i\in[n],i\neq j},\\ & (a_i)_{i\in S} \subseteq \{0,1\}^{|S|}, (y_i)_{i=0}^{n} \subseteq [0,1]^n) : \text{there exists a PSNE in}\\ & \mathbb{G} \text{ with the players in } S \text{ play according to } (a_i)_{i\in S}\\ & \text{and the attacker plays } (y_i)_{i=0}^{n} \text{ such that } \sum_{i=0}^{n} y_i = 1 \;\}.\end{aligned}$$

We will reduce our problem from Monotone 1 in 3-SAT where each clause of the 3-SAT has exactly three variables and consists of (un-negated) variables. We use the term variable(s) by default for un-negated variable(s), unless stated otherwise. The solution to the Monotone 1 in 3-SAT is to find a satisfiable assignment such that exactly one variable is true in each clause. The Monotone 1 in 3-SAT is known to be NP-complete [Garey and Johnson, 1979]. We denote the instances with satisfiable solutions as

$$\begin{aligned}\text{M 1 in 3-}SAT \;=\; & \{\; ((x_i)_{i\in[m]}, \wedge_{i=1}^{c} C_i, C_i = (\vee_{j=1}^{3} x_{i_j})) : \text{there exists a}\\ & \text{satisfiable assignment with exactly one}\\ & \text{variable true in each clause }\},\end{aligned}$$

where there are m variables, c clauses, and each clause has three (un-negated) variables. A satisfiable assignment is defined to be an assignment of all variables i to zero or one, $x_i \in \{0,1\}$, such that the boolean formula $\wedge_{i=1}^{c} C_i$ is true or satisfied (i.e., each clause C_i is true or satisfied and has exactly one variable true).

Below, given an instance of Monotone 1 in 3-SAT

$$\gamma = \left((x_i)_{i\in[m]}, \wedge_{i=1}^{c} C_i, C_i = (\vee_{j=1}^{3} x_{i_j})\right),$$

we are going to construct an instance of sites-games with partial assignments

$$\begin{aligned}\beta =& ([n], (C_i)_{i\in[n]}, (\alpha_i)_{i\in[n]}, (L_i)_{i\in[n]}, (\widehat{p}_i)_{i\in[n]}, (\widehat{q}_{ji})_{j,i\in[n],i\neq j},\\ & (a_i)_{i\in S} \subseteq \{0,1\}^{|S|}, (y_i)_{i=0}^{n} \subseteq [0,1]^n \text{ such that } \sum_{i=0}^{n} y_i = 1),\end{aligned}$$

that correspond to γ.

- There are $n = 2c + m$ players: two players for each clause and a player for each variable. The clause players and the variable players are indexed from 1 to $2c$ and $2c + 1$ to $2c + m$, respectively.

- First, we find $1 > L'' > C'' > 0$ and $1 > \widehat{p}'' > \frac{C''}{L''}$ such that $0 < \frac{C''}{L''\widehat{p}''} < 1$. Next, we find $\widehat{q} \in [0,1]$ such that $0 < \widehat{q} < \min\{\frac{L''\widehat{p}''}{3C''}, 1\}$. For completeness, we find $1 > \alpha'' > 0$. For each variable player $i \in \{2c+1, ..., 2c+m\}$, let $C_i = C''$, $\alpha_i = \alpha''$, $L_i = L'$, $\widehat{p}_i = \widehat{p}''$, and $y_i = \frac{C_i}{L_i \widehat{p}_i}$.

  The variable players are indifferent from playing the action invest or not invest.

- Next, using the values of the parameters defined above, we find $0 < C < L < 1$, $1 > \widehat{p} > \frac{C}{L} > 0$, $0 < y < \frac{C}{L\widehat{p}}$, and $1 > \alpha > 0$ such that $\frac{3C''\widehat{q}}{L''\widehat{p}''} > \frac{1}{1-\alpha}\left(\frac{C}{L} - y\widehat{p}\right) > \frac{2C''\widehat{q}}{L''\widehat{p}''}$. Indeed, such value is alway possible as we can make α and y to be arbitrarily small so that $\frac{1}{1-\alpha}\left(\frac{C}{L} - y\widehat{p}\right) \approx \frac{C}{L}$.

  For each clause player $i \in [c]$ such that $C_i = (\vee_{j=1}^{3} x_{i_j})$, $q_{(i_j+2c)i} = \widehat{q}$ for all j. To set the remaining parameters, for each clause player $i \in [c]$, set $C_i = C$, $L_i = L$, $\alpha_i = \alpha$, $p_i = p$, and $y_i = y$.

- Then, using the same values of the parameters defined for the variable players, we find $0 < C' < L' < 1$, $1 > \widehat{p}' > \frac{C'}{L'} > 0$, $0 < y' < \frac{C'}{L'\widehat{p}'}$, and $1 > \alpha' > 0$ such that $\frac{2C''\widehat{q}}{L''\widehat{p}''} > \frac{1}{1-\alpha'}\left(\frac{C'}{L'} - y'\widehat{p}'\right) > \frac{C''\widehat{q}}{L''\widehat{p}''}$.

  For each clause player $i \in \{c+1, ..., 2c\}$ such that $C_{i-c} = \left(\vee_{j=1}^{3} x_{(i-c)_j}\right)$, $q_{((c-i)_j+2c)i} = q$ for all j. To set the remaining parameters, for each clause player $i \in \{c+1, ..., 2c\}$, set $C_i = C'$, $L_i = L'$, $\alpha_i = \alpha'$, $p_i = p'$, and $y_i = y'$.

- Here, we construct a partial action profile for some of the players. In particular, for each clause player $i \in [c]$, $a_i = 0$ and $a_{i+c} = 1$. Thus, we are giving a partial action profile of all clause players. For completness, let $y_0 = 1 - \sum_{i=1}^{n} y_i$.

Lemma 14. *$\gamma \in$ M 1 in 3-SAT $\implies \beta \in$ sites-game.*

Proof. Given a satisfiable assignment for γ, we show how to construct a PSNE for β. Let $x^{(1)} = \{i \in [m] : x_i = 1\}$ be the indices of the variables that are assigned a value of one in the satisfiable assignment. For consistence, we let a_i to denote the action of any player $i \in [n]$ and construct a PSNE as follows. For each of the variable player $i \in \{2c+1, ..., 2c+m\}$, $a_i = 1$ if $(i-2c) \in x^{(1)}$

and $a_i = 0$ otherwise. Together with the partial action profile of the clauses, we will call this constructed pure-strategy profile $a = (a_1, ..., a_n)$.

To show that a is a PSNE, we argue that each player is playing its best-response. First, we consider the clause players. Recall that best-response correspondence of a clause player $i \in [c]$ is

$$\mathcal{BR}_i(x_{\text{Pa}(i)}, y_{\text{PF}(i)}) \equiv \begin{cases} \{1\}, & \text{if } \widehat{s}_i(x_{\text{Pa}(i)}, y_{\text{PF}(i)}) > \widehat{\Delta}_i, \\ \{0\}, & \text{if } \widehat{s}_i(x_{\text{Pa}(i)}, y_{\text{PF}(i)}) < \widehat{\Delta}_i, \\ [0,1], & \text{if } \widehat{s}_i(x_{\text{Pa}(i)}, y_{\text{PF}(i)}) = \widehat{\Delta}_i. \end{cases}$$

where $\widehat{\Delta}_i \equiv \frac{C_i}{L_i \widehat{p}_i}$, $\widehat{s}_i(x_{\text{Pa}(i)}, y_{\text{PF}(i)}) \equiv y_i + \frac{1-\alpha_i}{\widehat{p}_i} r_i(x_{\text{Pa}(i)}, y_{\text{Pa}(i)})$. Notice that, to determine the best-response strategy of player i, without loss of generality, we can also compare the values of $\frac{1}{1-\alpha_i}\left(\frac{C_i}{L_i} - y_i \widehat{p}_i\right)$ and $r_i(x_{\text{Pa}(i)}, y_{\text{Pa}(i)})$. By our construction, $Pa(i) = \{i_1, i_2, i_3\}$ (which corresponds to variables $x_{i_1}, x_{i_2}, x_{i_3}$ of clause i) and $r_i(x_{\text{Pa}(i)}, y_{\text{Pa}(i)}) = \sum_{j \in \text{Pa}(i)} \frac{C''}{L''\widehat{p}}(1 - x_j)\widehat{q}$.

Moreover, by the satisfiable assignment, exactly one variable in $\text{Pa}(i)$ is assigned to a value of one which corresponds to exactly one variable player that plays action one. Therefore, $r_i(x_{\text{Pa}(i)}, y_{\text{Pa}(i)}) = \frac{2C''\widehat{q}}{L''\widehat{p}}$. By our construction, $\frac{3C''\widehat{q}}{L''\widehat{p}''} > \frac{1}{1-\alpha_i}\left(\frac{C_i}{L_i} - y_i\widehat{p}_i\right) > \frac{2C''\widehat{q}}{L''\widehat{p}''}$. It follows that $r_i(x_{\text{Pa}(i)}, y_{\text{Pa}(i)}) < \frac{1}{1-\alpha_i}\left(\frac{C_i}{L_i} - y_i\widehat{p}_i\right)$, and the i's best-response is zero. This holds for all clause players $i \in [c]$. On the other hand, for the clause player $i \in \{c+1, ..., 2c\}$, $r_i(x_{\text{Pa}(i)}, y_{\text{Pa}(i)}) = \frac{2C''\widehat{q}}{L''\widehat{p}}$ as well. By our construction, $\frac{2C''\widehat{q}}{L''\widehat{p}''} > \frac{1}{1-\alpha_i}\left(\frac{C_i}{L_i} - y_i\widehat{p}_i\right) > \frac{C''\widehat{q}}{L''\widehat{p}''}$, it follows that $\frac{1}{1-\alpha_i}\left(\frac{C_i}{L_i} - y_i\widehat{p}_i\right) < r_i(x_{\text{Pa}(i)}, y_{\text{Pa}(i)})$ and $a_i = 1$ is the best-response.

For each variable player $i \in \{2c+1, ..., 2c+m\}$, i has no parent and i's overall risk is 0. To determine whether i plays the action invest or not invest, we only need to compare the value of $\frac{C_i}{L_i}$ and $y_i\widehat{p}_i$. By construction, $\frac{C_i}{L_i\widehat{p}_i} = y_i$ for all variable players i, we have that the variable players are indifferent between playing one and zero. Hence, the pure-strategy profile a is a PSNE. □

Lemma 15. *$\beta \in$ sites-game $\implies \gamma \in$ M 1 in 3-SAT.*

Proof. Now we show how to construct a satisfiable assignment for γ given a PSNE of β. Let $a = (a_1, ..., a_n)$ be a PSNE of β. For each variable $i \in [m]$, if $a_{2m+i} = 1$ then $x_i = 1$ and if $a_{2m+i} = 0$ then $x_i = 0$. To show that each clause, say $i \in [c]$, has exactly one variable that is true, we observe the best-response of clause players i and $c+i$ that correspond to clause i. Given the fixed action of $a_i = 0$ and $a_{c+i} = 1$ at a PSNE, it implies that $r_i(x_{\text{Pa}(i)}, y_{\text{Pa}(i)}) < \frac{1}{1-\alpha_i}\left(\frac{C_i}{L_i} - y_i\widehat{p}_i\right)$ and $r_{c+i}(x_{\text{Pa}(c+i)}, y_{\text{Pa}(c+i)}) > \frac{1}{1-\alpha_{c+i}}\left(\frac{C_{c+i}}{L_{c+i}} - y_{c+i}\widehat{p}_{c+i}\right)$. Since $\frac{3C''\widehat{q}}{L''\widehat{p}''} > \frac{1}{1-\alpha_i}\left(\frac{C_i}{L_i} - y_i\widehat{p}_i\right) > \frac{2C''\widehat{q}}{L''\widehat{p}''}$, $\frac{2C''\widehat{q}}{L''\widehat{p}''} > \frac{1}{1-\alpha_{c+i}}\left(\frac{C_{c+i}}{L_{c+i}} - y_{c+i}\widehat{p}_{c+i}\right) > \frac{C''\widehat{q}}{L''\widehat{p}''}$, $\text{Pa}(c+i) = \text{Pa}(i)$, $|Pa(i)| = 3$, and the transfer risks are the same, we have $s_{c+i}(a_{\text{Pa}(c+i)}) = \frac{2C''\widehat{q}}{L''\widehat{p}''}$. This implies that exactly one of the variables is true. □

It is easy to see that given a (partial) pure-strategy profile, we can verify whether it is a PSNE of a sites-game in polynomial time. This fact, together with Lemma 14 and Lemma 15, we have our hardness result. □

Worst case, we need to consider the y just described, should other strategies fail to be a part of any MSNE. Another challenge is that even if we can compute all exact MSNE, there could be exponentially many of them to check. In the next section, we look for efficient algorithms to compute an approximate MSNE in various graph structures.

5.2 FPTAS to Compute ϵ-MSNE of Tree-like IDD Games

In this section, we compute ϵ-MSNE in a subclass of IDD games. In particular, we study different graph structures among the sites. We note that the attacker is connected to all of the sites even if we do not point it out explicitly.

We have been using the notion of ϵ-MSNE in the pervious chapter without formally defined it. Below, we provide a more formal definition.

Definition 15. *A mixed-strategy (x^*, y^*) is an ϵ-MSNE of an IDD game if (1) for all $i \in [n]$, $M_i(x_i^*, x_{Pa(i)}^*, y_i^*, y_{Pa(i)}^*) \leq \min_{x_i} M_i(x_i, x_{Pa(i)}^*, y_i^*, y_{Pa(i)}^*) + \epsilon$, and (2) $U(x^*, y^*) \geq \max_y U(x^*, y) - \epsilon$.*

An exact MSNE $\equiv$ 0-MSNE. Moreover, we assume that all the cost and utility functions are individually normalized to $[0,1]$ and $\epsilon \in [0,1]$; otherwise ϵ is not well-defined.

We will start off simple by considering a *directed star (DS)* graph structure. We show that there is a *fully polynomial-time approximation scheme (FPTAS)* to compute an ϵ-MSNE in DS-IDD games. Roughly speaking, an FPTAS's running time is some polynomial of the input and $\frac{1}{\epsilon}$ for $1 < \epsilon < 1$ [Vazirani, 2001]. Then we generalize the result to *directed trees (DT)*. Despite the simplicity of the graphs, one can envision very important real-world applications such as protection of supply chains and other hierarchical structures (e.g. see Agiwal and Mohtadi [2008]).

5.2.1 Directed Stars

Let the source node correspond to player n, and the remaining $n - 1$ sink nodes correspond to players' $1, \ldots, n-1$. The directed star (DS) is equivalent to a directed tree with a single root at n with $n - 1$ leaves and no internal nodes.

Since the domain of the variables (i.e., mixed strategies) is $[0, 1]$, a direct optimization method to compute an MSNE would require solving a highly non-linear optimization problem: cubic objective function for the attacker with quartic constraints for the sites. *An alternative is to discretize the continuous space of the x_i's and y_i's.*

Let $\mathcal{X} \equiv \mathcal{X}(\Delta_x) \equiv \{0, \tau_x, 2\tau_x, \ldots, (\Delta_x - 1)\tau_x, 1\}$ and $\mathcal{Y} \equiv \mathcal{Y}(\Delta_y) \equiv \{0, \tau_y, 2\tau_y, \ldots, (\Delta_y - 1)\tau_y, 1\}$ be the respective *discretization* of the interval $[0, 1]$ for the sites and the attacker, where $\tau_x \equiv \frac{1}{\Delta_x}$ and $\tau_y \equiv \frac{1}{\Delta_y}$ are the respective *discretization sizes*, and Δ_x and Δ_y are the respective *discretization lengths*. The discretization defines the domains of x_i and y_i to be $\mathcal{X}$ and $\mathcal{Y}$, respectively. Moreover, $|\mathcal{X}| = \Delta_x$ and $|\mathcal{Y}| = \Delta_y$. Of course, there is an extra constraint for the y_i's in $\mathcal{Y}$: $\sum_{i=1}^{n} y_i \leq 1$ for $y \in \mathcal{Y}^n$. We will determine the values of Δ_x and Δ_y to guarantee an ϵ-MSNE later in the section, but for now, assume they are given. A simple brute-force algorithm to compute an ϵ-MSNE is to check all possible discrete combinations and would take, $O\left((\frac{1}{\Delta_x}\frac{1}{\Delta_y})^n\right)$ time, to run in the worst case.

Indeed, we can apply the principle of *dynamic programming* [Bellman, 2003] and design an efficient algorithm to compute ϵ-MSNE that is provably an FPTAS. The key idea is to realize that given a strategy (x_n, y_n) of the root n, the leaves' decisions are independent of each other. However, there is a sum less than or equal to one constraint for the attacker (i.e., $\sum_{i=1}^{n} y_i \leq 1$). Indeed, for each possible combination of (x_n, y_n), we can run a dynamic programming algorithm (to be presented later) based on some ordering of the nodes and obtain a (best) value for each (x_n, y_n). Clearly, the best (x_n^*, y_n^*) that obtains the maximum value among all other (x_n, y_n)'s is the best possible strategy for the attacker. This guarantees that the attacker would not deviate to a different strategy. Moreover, the dynamic programming algorithm would produce solutions that ensure the leave players are best-responding. More formally, we define the following mathematical expressions for the dynamic programming algorithm. This will give us an FPTAS for DS-IDD games.

Upstream pass: Collection of conditional ϵ-MSNE computation. First, we impose an ordering on the leaves, that is, we order the leaves in increasing order. Let $\overline{M}_i(x_i, y_i, x_n, y_n) \equiv M_i(x_i, y_i, x_n, y_n) - x_i C_i - y_i C_i^0$ be the attacker's utility for attacking i. For each leaf $i = 1, \ldots, n-1$, we compute the set of individual conditional tables (in this order),

$$\begin{aligned}
&\overline{T}_{i,n}(x_n, y_n, v_i, x_i, y_i, v_{i-1}) \equiv \\
&\overline{M}_i(x_i, y_i, x_n, y_n) + \\
&\log\left(\mathbb{1}[v_i = y_i + v_{i-1}]\right) + \\
&\log\left(\mathbb{1}\left[x_i \in \mathcal{BR}^{\epsilon}_{x_i}(y_i, x_n, y_n)\right]\right) + \\
&T_{i-1,n}(x_n, y_n, v_{i-1})
\end{aligned}$$

$$T_{i,n}(x_n, y_n, v_i) \equiv \max_{(x_i, y_i, v_{i-1})} \overline{T}_{i,n}(x_n, y_n, v_i, x_i, y_i, v_{i-1})$$

$$W_{i,n}(x_n, y_n, v_i) \equiv \arg\max_{(x_i, y_i, v_{i-1})} \overline{T}_{i,n}(x_n, y_n, v_i, x_i, y_i, v_{i-1})$$

where $T_{0,n}(x_n, y_n, s_0) = 0$ for all (x_n, y_n, s_{i_0}). Each $T_{i,n}$ specifies the maximum possible utility an attacker can get by attacking all the leaves up to i given that the attacker will attack the root n with probability y_n, the root n to invest with probability x_n, and the allowable remaining probability of an attack v_i. The first and the second log-terms are to ensure that the overall probability of attack does not exceed the allowable limit and player i is playing best-respond strategies, respectively. This is similar to the DL case. Computing each "table of sets" T's and W's, given above, all take $O(\Delta_x^2 \Delta_y^4)$

each. For n, the *root* of the tree, we compute

$$\begin{aligned}\overline{R}_0(s_0, x_n, y_n, s_n) \equiv & \overline{M}_n(x_n, y_n) + \\ & \log\left(\mathbb{1}[s_0 = s_n + y_n]\right) + \\ & \log\left(\mathbb{1}[x_n \in \mathcal{BR}_n^{\epsilon}(y_n)]\right) + \\ & R_n(x_n, y_n, s_n)\end{aligned}$$

$$R_0(s_0) \equiv \max_{(x_n, y_n, s_n)} \overline{R}_0(s_0, x_n, y_n, s_n)$$

$$W_0(s_0) \equiv \arg\max_{(x_n, y_n, s_n)} \overline{R}_0(s_0, x_n, y_n, s_n)$$

Clearly, computing R_0 and W_0 takes $O(\Delta_x \Delta_y^3)$. As mentioned earlier, for each combination of (x_n, y_n), we are going to compute the best value an attacker can obtain. The computation of R_0 does exactly this.

Downstream pass: Assignment phase. The assignment phase is essentially the backtracking phrase in the dynamic programming algorithm where we follow the "back pointers" to find the mixed-strategies for the players and the attacker. For the "downstream" or assignment pass, we are going to start with the root and find $s_0^* \in \arg\max_{s_0} R_0(s_0)$. Because of the discretization result of Theorem 14, there always exists an ϵ-MSNE, and thus, there is a s_0^* such that $R_0(s_0^*) < -\infty$. We set the mixed-strategy of the root to be some $(x_n^*, y_n^*, s_n^*) \in W_0(s_0^*)$. Starting from the opposite order of upstream pass (i.e., $n-1, ..., 1$), we set the mixed-strategies of the leaves according to $v_{n-1}^* \leftarrow s_n^*$, and for $i = n-1, \ldots, 1$,

$$(x_i^*, y_i^*, s_i^*, v_{i-1}^*) \in W_i(x_n^*, y_n^*, v_i^*) \ .$$

By construction the resulting (x^*, y^*) is an ϵ-MSNE of the DS-IDD game.

The key to show that this dynamic-programming algorithm produces an ϵ-MSNE for the DS-IDD games is the discretization sizes. The question is, how small can we make Δ_x and Δ_y and still guarantee an ϵ-MSNE in the discretized space? A more general result about sparse discretization for graphical games [Ortiz, 2014] provides the answer. Below, we formally state the result of Ortiz [2014] for graphical games.

Theorem 14. *[Ortiz, 2014] For anym-action graphical game and any $\epsilon > 0$, a (individually-uniform) discretization with [discretization size]*

$$s_i = \left\lceil \frac{2|A_i| \max_{j \in Pa(i) \cup Ch(i)} |Pa(j) \cup Ch(j)|}{\epsilon} \right\rceil$$

for each player i is sufficient to guarantee that for every true (i.e., not approximate) MSNE of the game, its closest mixed-strategy profile in the induced discretized space is also an ϵ-MSNE of the game.

In other words, to get an ϵ-MSNE, we need to set the discretization sizes as specified above for each player in the game.

Lemma 16. *Let $\Delta_x = O(\frac{4n}{\epsilon})$ and $\Delta_y = O(\frac{2n^2}{\epsilon})$. There is a dynamic-programming algorithm that computes an ϵ-MSNE of DL-IDD games in time $O(n(\Delta_x \Delta_y^2)^2) = O(\frac{n^{11}}{\epsilon^6})$.*

Proof. From Theorem 14, we need to set the appropriate discretization sizes for the sites and the attacker. For each site i, $|A_i| = 2$ because i has only two actions, and $\max_{j \in \mathrm{Pa}(i) \cup \mathrm{Ch}(i)} |\mathrm{Pa}(j) \cup \mathrm{Ch}(j)| = n$ because the attacker is connected to all of the sites. Thus, we have $\Delta_x = O(\frac{4n}{\epsilon})$. There are $n + 1$ actions for the attacker (including no attack). Since the attacker has the root node n as a neighbor, $|\mathrm{Pa}(n) \cup \mathrm{Ch}(n)| = n$. Therefore, $\Delta_y = O(\frac{2n^2}{\epsilon})$. Moreover, the dynamic-programming has size at most $O(\Delta_x^2 \Delta^4)$ and takes at most $O(n(\Delta_x \Delta_y^2)^2)$ to run. A simply substitution gives us the claimed running times. □

Our next corollary follows from the above lemma and the definition of FPTAS.

Corollary 2. *There is an FPTAS to compute an ϵ-MSNE in DS-IDD games.*

5.2.2 Directed Trees

We now generalize the last result even further to arbitrary DT-IDD games, yielding one of our main technical results.

Theorem 15. *Let $\Delta_x = O(\frac{4n}{\epsilon})$ and $\Delta_y = O(\frac{2n^2}{\epsilon})$. There is a dynamic-programming algorithm that computes an ϵ-MSNE of DL-IDD games in time $O(n(\Delta_x \Delta_y^2)^2) = O(\frac{n^{11}}{\epsilon^6})$.*

Proof. Let n denote a site/node in the directed tree with a single source (i.e., the root of the tree). Let $(i_1, \ldots, i_{k_n})$ be a sequence ordering the set of children of n, $\mathrm{Ch}(n) \equiv \{i_1, \ldots, i_{k_n}\}$, where $k_n \equiv |\mathrm{Ch}(n)|$. The following conditions expresses the dynamic programming corresponding to the "upstream pass" of the algorithm. For all n, *except the root* of the directed tree, we (recursively) define

$$R_n(x_n, y_n, s_n) \equiv T_{i_{k_n},n}(x_n, y_n, s_n)$$

such that, for all $j = 1, \ldots, k_n$, we define

$$\begin{aligned} T_{i_j,n}(x_n, y_n, v_{i_j}) \equiv \max_{(x_{i_j}, y_{i_j}, s_{i_j}, v_{i_{j-1}})} & \overline{M}_{i_j}(x_{i_j}, y_{i_j}, x_n, y_n) \\ & + \log\left(\mathbb{1}\left[v_{i_j} = s_{i_j} + y_{i_j} + v_{i_{j-1}}\right]\right) \\ & + \log\left(\mathbb{1}\left[x_{i_j} \in \mathcal{BR}^{\epsilon}_{x_{i_j}}(y_{i_j}, x_n, y_n)\right]\right) \\ & + R_{i_j}(x_{i_j}, y_{i_j}, s_{i_j}) \\ & + T_{i_{j-1},n}(x_n, y_n, v_{i_{j-1}}) \ , \end{aligned}$$

$W_{i_j,n}(x_n, y_n, v_{i_j})$ is the arg max of the same optimization (i.e., the set of "witnesses" containing the values of $(x_{i_j}, y_{i_j}, s_{i_j}, v_{i_j-1})$ that achieve the maximum values of the optimization given each (x_n, y_n, v_{i_j})), and, to simplify the presentation, we use the boundary conditions (1) $T_{i_0,n}(x_n, y_n, s_{i_0}) = 0$ for all (x_n, y_n, s_{i_0}); and (2) if i_j is a *leaf* of the tree, then $R_{i_j}(x_{i_j}, y_{i_j}, s_{i_j}) = 0$ for all $(x_{i_j}, y_{i_j}, s_{i_j})$. If n is the *root* of the tree, we compute

$$\begin{aligned} R_0(s_0) \equiv \max_{(x_n, y_n, s_n)} & \overline{M}_n(x_n, y_n) \\ & + \log\left(\mathbb{1}[s_0 = s_n + y_n]\right) \\ & + \log\left(\mathbb{1}[x_n \in \mathcal{BR}^{\epsilon}_n(y_n)]\right) \\ & + R_n(x_n, y_n, s_n) \ , \text{ and} \end{aligned}$$

$W_0(s_0)$ is the arg max of the same optimization (i.e., the set of "witnesses" containing the values of (x_n, y_n, s_n) that achieve the maximum values of the optimization given each s_0 in the discretized grid of probability values in $[0, 1]$).

For the "downstream" or assignment pass, first find $s_0^* \in \arg\max_{s_0} R_0(s_0)$. Note that such s_0^* with $R_0(s_0^*) < -\infty$ because of the properties of the discretization and the existence of MSNE. Set the values of the root node, denoted by n, to some $(x_n^*, y_n^*, s_n^*) \in W_0(s_0^*)$. Then (recursively) set the values of the children of n, in the reversed order in which the the dynamic program computes the maximizations: set $v^*_{i_{k_n}} \leftarrow s^*_n$, and for $j = k_n, \ldots, 1$.

$$(x^*_{i_j}, y^*_{i_j}, s^*_{i_j}, v^*_{i_{j-1}}) \in W_{i_j}(x^*_n, y^*_n, v^*_{i_j}) \ .$$

We repeat the same assignment process for all of the nodes in the tree. Recall that there will always be at least one witness value during the assignment phase because of the properties of the discretization and the existence of MSNE. By construction (i.e., the properties of dynamic programming and the discretization of Theorem 14), the resulting (x^*, y^*) is an ϵ-MSNE of the DT-IDD game. The running time would be $O(n\Delta_x^2\Delta_y^4)$. Our result follows by applying the same analysis of Lemma 16.

□

Corollary 3. *There is an FPTAS to compute an ϵ-MSNE in DT-IDD games.*

Note that our results are nontrivial within the context of the state-of-the-art in computational game theory. We are working a graph structure where there is one node (the attacker) connecting to *all* the nodes of the tree (the sites). Naively applying the traditional well-known dynamic programming algorithms by Kearns et al. [2001] and Elkind et al. [2006] to our problem would not give us any FPTAS. In fact, their game representation size is exponential in the number of neighbors instead of our *linear* representation size. Moreover, finding ϵ-MSNE in general degree-3 graphical games is PPAD-hard Elkind et al. [2006]. Our IDD games have more than 3 degrees. In fact, because the attacker is connected to all the nodes in the network, as a graphical game with normal-form representation of the local payoff matrices, the graph of the IDD games is completely connected (i.e., the attacker's mixed strategy imposes a global constraint). But the local payoff functions in our case are compactly representable in parametric form. Still, we provide an FPTAS to compute ϵ-MSNE in interesting subclasses of IDD games.

5.3 A Heuristic to Compute ϵ-MSNE

In this section, we introduce a heuristic to compute ϵ-MSNE on *arbitrary* graphs. Previously, we show that showed that *best-response-gradient dynamics (BRGD)* [Fudenberg and Levine, 1998, Nisan et al., 2007, Shoham and Leyton-Brown, 2009] can efficiently solve *Internet games (IGs)*, and can output ϵ-MSNE up to $\epsilon = 0.001$. Recall that BRGD begins by initializing x_i

and y_i in $[0, 1]$ for all sites i such that $\sum_{i=1}^{n} y_i \leq 1$. At each round, BRGD update $x_i \leftarrow x_i - 10 * (M_i(1, y_i, x_{\text{Pa}(i)}, y_{\text{Pa}(i)}) - M_i(0, y_i, x_{\text{Pa}(i)}, y_{\text{Pa}(i)})$ and $y_i \leftarrow y_i + 10 * (U_i(x) - U(x, y))$, where the M_i's and U functions are normalized to $[0, 1]$ and we use 10 as the learning-rate/step-size in our case. Here, we evaluate our heuristic using IGs randomly generated according above.

First we look at the attacker's behavior at an ϵ-MSNE. We generate a few IG instances and run BRGD until it converges to an ϵ-MSNE for $\epsilon \in \{0.001, 0.002, \ldots, 0.009\}$. We observe that in a 0.001-MSNE, (1) there is a positive, almost-deterministic correlation between the probability of an attack and the utility the attacker obtained from attacking the sites and (2) the attacker always target the sites with the highest potential utility (i.e., the maximum utility the attacker can get by attacking the sites with probability 1). This observation is consistent with other IGs and holds across the different ϵ-MSNE for various ϵ values. Figure 5.1 shows evidence of this behavior. Indeed, the main take away is that the attacker tends to favor (or target) sites with highest expected utility. As observed, the attack seems to have some distributional form.

In what follows, we assume that the attacker is using *smoothed-best-response* [Fudenberg and Levine, 1998] and the attack distribution has the form of a Gibbs-Boltzmann distribution. Although what follows is a relatively standard derivation and relatively common across many communities by now, we still provide the derivation for completeness.

Let *(Shannon's) entropy function* be

$$H(\mathbf{y}) \equiv \sum_{i=0}^{n} y_i \ln \frac{1}{y_i}$$

$$\text{s.t.} \sum_{i=0}^{n} y_i = 1$$

where y_0 denotes the probability of no attack, $y_0 = 1 - \sum_{i=1}^{n} y_i$. As before we assume that U is normalized to $[0, 1]$. Given $x \in [0, 1]^n$, we can compute y such that it maximizes the attacker's utility by solving the following maximization:

$$\max_{\mathbf{y} \geq 0} U(x, y) + cH(y)$$

$$\text{s.t.} \sum_{i=0}^{n} y_i = 1$$

for some real-valued constant $c > 0$. As is standard for this type of problems, the corresponding Lagrangian function and its first partial derivative with respect to y_i are

$$L(y, \lambda) \equiv U(x, y) + cH(y) + \lambda \left(1 - \sum_{i=0}^{n} y_i \right) \text{ and}$$
$$\frac{\partial L(y, \lambda)}{\partial y_i} = U_i(x) - c(\ln y_i + 1) - \lambda \; .$$

Solving for y_i, we have $y_i = \exp(\frac{U_i(x) - \lambda - c}{c}) \propto \exp(\frac{U_i(x)}{c})$. Hence the optimal y^* is unique and its individual components

$$y_i^* \equiv \frac{\exp(U_i(x_i, x_{\mathrm{Ch}(i)}))/c)}{\sum_{i=0}^{n} \exp(U_i(x_i, x_{\mathrm{Ch}(i)}))/c)} \; .$$

The interpretation of c is that it controls the *precision* of the attacker and make the utility more distinct. The parameter c is really the precision or *temperature parameter* of the Gibbs-Boltzmann distribution: increasing c leads to the uniform distribution, while decreasing c produces ϵ-MSNE with lower ϵ because c restricts the effect of the entropic term in that case. In fact, at temperature $c = 0$, we recover the original best-response for the attacker.

This form for the attacker's mixed strategy $\mathbf{y}$ has several attractive properties: (1) sites with high utility will have higher probability of an attack and (2) the respective expected utility and the probability of an attack are positively correlated (higher probability of attack implies higher expected utility gain). We observe these characteristics in our experiments (Figure 5.1).

Based on the previous discussion, we propose the following heuristic to compute ϵ-MSNE. The heuristic starts by initializing all of the sites investment level x_i to 0. It then updates the probability of attack for each site and increments the investment level of the site by a small amount (currently 0.001) for sites that do not satisfy the following condition: $R_i \geq y_i\hat{p}_i + (1 - \alpha_i)\sum_{j \in \mathrm{Pa}(i)} y_j(1 - x_j)\hat{q}_{ji}$. The algorithm terminates either when all of the sites satisfy the condition or when it reaches the maximum number of iterations. The condition, $R_i \geq y_i\hat{p}_i + (1 - \alpha_i)\sum_{j \in \mathrm{Pa}(i)} y_j(1 - x_j)\hat{q}_{ji}$, for site i is the threshold for i to not invest. A nice property of this is that given the attacker's Gibbs-Boltzmann distribution, for any site i, given the strategies of others, the attack decreases monotonically with x_i. As a result, no site has an incentive to increase its investment to violate the constraint

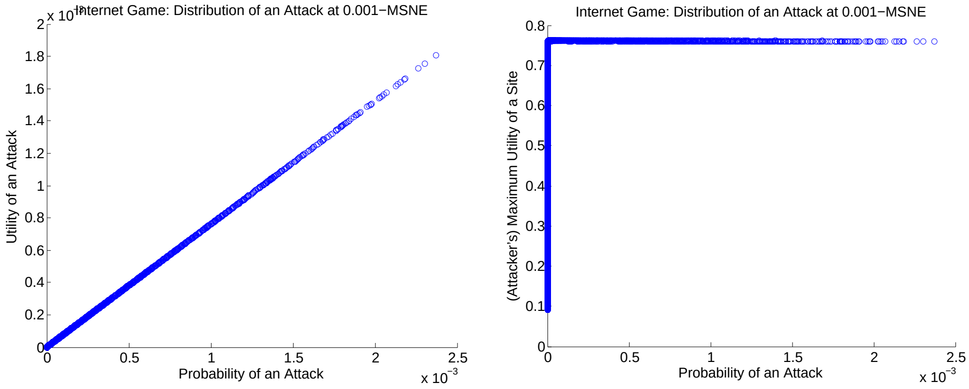


Figure 5.1: **Attack Distribution of Internet Game.**

above. Consequently, to justify the use of the condition in our heuristic in IGs, we observe that in all of the IGs we generated, the percentage of the sites at the 0.001-MSNE we obtained that satisfies the above condition is $\geq 98\%$. The quality of an ϵ-MSNE obtained by our heuristic depends on the percentage of the sites that satisfy the condition at an ϵ-MSNE. Note that if a high percentage of the sites do not satisfy the condition at the ϵ-MSNE, we can reverse the heuristic by initializing all of the sites investment level x_i to 1 and lower the x_i's until all sites satisfy the opposite constraint.

Algorithm 8 provides pseudocode for the resulting attacker-smoothed-best-response heuristic to compute an approximate MSNE in arbitrary IDD games as discussed.

5.3.1 Evaluation of Heuristic on Internet Games

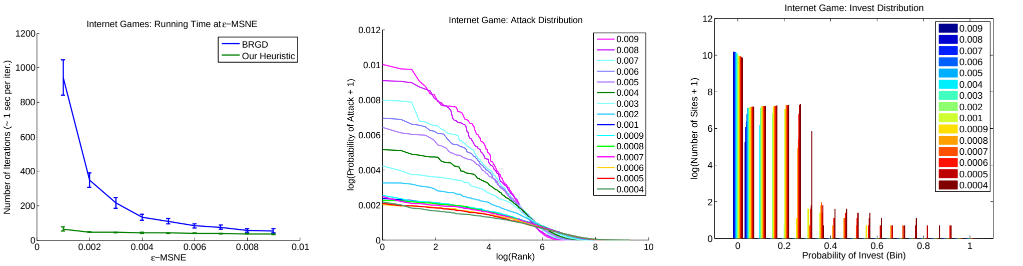
To evaluate our heuristic, we randomly generated ten IGs and compare the results to those obtained using BRGD.

The first question we address is, what is the relation between the constant c and the actual approximation quality ϵ achieved in practice? Table 5.1 shows the impact c has on ϵ, for the smallest ϵ-MSNE we can obtain for an instance of the IGs (others are similar). Take-home message: ϵ deceases with c as expected. For the remaining of this section, we will fix $c = 0.001$ when comparing to BRGD as BRGD cannot find ϵ-MSNE beyond 0.0009-MSNE within 10,000 iterations (1 sec. per iteration).

```
Algorithm 8: Compute an $\epsilon$-MSNE in IDD Games
Input  : An instance of an $n$-player IDD game, $T_{\max}$
Output: $(\mathbf{x}, \mathbf{y})$ - An $\epsilon$-MSNE
Let $x_i \leftarrow 0$ for all $i = 1, 2, ..., n$
Let iteration $\leftarrow 0$
Let increment $\leftarrow 0.001$
Let Converge $\leftarrow$ false
while  not Converge AND iteration $< T_{\max}$ do
    Converge = true
    $\bar{y}_i \leftarrow \exp(\frac{U_i(\mathbf{x})}{c})$ for all $i = 1, 2, ..., n$
    $y_i = \frac{\bar{y}_i}{\sum_{i=0}^{n} \bar{y}_i}$ for all $i = 1, 2, ..., n$
    foreach $i = 1, 2, ...n$ do
        if $R_i < y_i \hat{p}_i + (1 - \alpha_i) r_i(\mathbf{x}_{Pa(i)}, \mathbf{y}_{Pa(i)})$ then
            $x_i = x_i +$ increment (if $x_i > 1, x_i = 0$)
            Converge = false
        end if
    end foreach
    iteration = iteration + 1
end while
```

c	smallest ϵ
0.05	0.06
0.01	0.008
0.005	0.004
0.001	0.0009
0.0005	0.0006
0.0001	0.0004

Table 5.1: **Selection of the constant c for our heuristic.**



(a) (b)

Figure 5.2: **Properties of our heuristic.**(a) BRGD vs. our heuristic running time; (b) Attacker's attack and sites' investment distribution on ϵ-MSNE

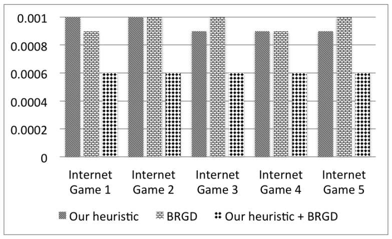


Figure 5.3: **Combing BRGD and our heuristic** Internet Games: BRGD Improvement (y-axis represents the ϵ values)

Comparing Running Time of BRGD and Our Proposed Heuristic

Next we study the time that the ten IG instances took to converge to an ϵ-MSNE using BRGD and our heuristic. We consider the running time in terms of the number of iterations the algorithm takes to achieve a particular ϵ-MSNE. Each iteration is roughly 1 sec. for both BRGD and our heuristic. Figure 5.2(a) shows that the running time of our heuristic is considerably faster than BRGD. The rate at which the number of iterations increases as ϵ decreases seems extreme for our heuristic—it is almost constant!—relative to that for BRGD. Not only is our heuristic faster than BRGD but it can also find ϵ-MSNE with smaller ϵ.

As an application, we could run our heuristic until it reaches an ϵ-MSNE or converges. Then use the output of our ϵ-MSNE to initialize BRGD. Figure 5.3 shows the relative improvement over our heuristic on some IGs. It improves our 0.001/0.0009-MSNE to 0.0006-MSNE.

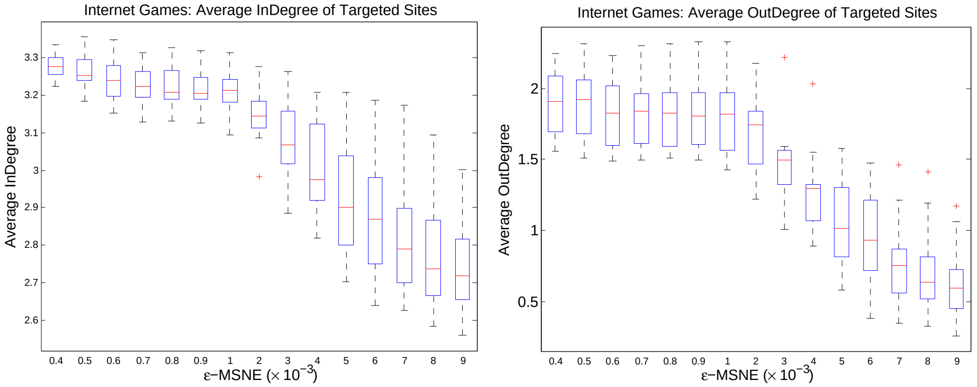


Figure 5.4: **Degrees of the targeted nodes at ϵ-MSNE** Internet Games: average indegree (top) and average outdegree (bottom) of the targeted sites over ϵ-MSNE

Attacker and Sites' Equilibrium Behavior

We study whether the same equilibrium behavior by the attacker and sites in our earlier section. continues as we lower ϵ. The following results are a direct output of our heuristic. Figure 5.2(b) shows the attack distribution (left) and the investment distribution (right) at ϵ-MSNE, for different ϵ values, on an IG instance. Our results are consistent with those of earlier section and persist for lower ϵ values. We see that as ϵ decreases, the attacker targets more sites while lowering the probability of the direct attack, and more sites move from not invest to partially invest.

Network Structure of an Attack

Next, we present experimental results on the average indegree and outdegree of the targeted sites at ϵ-MSNE to understand the "network structure of the attack" as in earlier section. Figure 5.4 shows exactly this. To summarize our experimental results, we can clearly observe that as ϵ decreases both the average indegree and outdegree increase. The results for lower ϵ values indicate the average indegree and outdegree are stabilizing and converging as ϵ decreases. This is also consistent with the observations made by earlier section. This consistency also adds evidence to the effectiveness of our proposed heuristic for very low ϵ values.

5.4 Conclusion

We study the problem of computing an ϵ-MSNE in IDD games. We show that determining whether an attacker's strategy can be a part of a MSNE is unlikely to have an efficient algorithm. However, there is an FPTAS to compute an ϵ-MSNE when the underlying game-graph is a DT. For general IDD games, we construct a heuristic that computes ϵ-MSNE in IGs effectively and efficiently. An open problem is to show that computing an MSNE in IDD games is PPAD-hard. Another open problem is to generalize the FPTAS for DT to directed acyclic graphs of bounded-width of some kind.

Chapter 6

Learning Game Parameters from MSNE: An Application to Learning the Generalized IDS Games

6.1 Introduction

A survey is an important tool for eliciting information from agents in large populations. Many government agencies such as the *Centers for Disease Control and Prevention (CDC)* in the United States sample some subsets of the population, and elicit information from them via a survey or a questionnaire. For example, the CDC can ask an agent, "Did you take the H1N1 vaccine this month?"

We are particularly interested in this type of question because it reveals the action an agent took previously. Indeed, if we believe that the agents are rational, self-interested, and their decisions affect the decisions of others, then we can attempt to learn and infer a game in which the actions the agents took are the "best-responses" of all other agents, and no agent would benefit from unilaterally deviating from their current action (i.e., from taking a vaccine to not taking a vaccine, or vice versa). These actions (collectively) are known as *pure-strategy Nash Equilibrium (PSNE)* and we will define it more formally later. However, quite often, we do not get to see the completed survey of each individual in the population. This is, perhaps, due to the large amount of data and information: it is impossible to keep track of all the details. Instead, what we typically can publicly obtain is some compact representation of the data that summarizes or aggregates the information collected.

More concretely, using the CDC vaccination data as our running example, we can observe the monthly state vaccination percentages (along with standard deviations) for different age groups, race groups, and types of vaccinations. These vaccination percentages represent the *average* behavior of the people in the USA. Indeed, if we view each state as an agent, we can interpret each vaccination rate at each state in the USA as the probability, or, using game-theory parlance, the "mixed-strategy", that the corresponding "state agent" vaccinates against a particular disease. (Please keep in mind that the state agent's mixed-strategy really corresponds to the percentage of people in that state that would decide to vaccinate against that particular disease.) Moreover, if we further consider the fact that the behaviors (vaccination rates) of the state agents affect the vaccination rates of others, then we can model the vaccination scenario, at the level of states, as a game. In particular, agents are the states in the USA, actions are either to vaccinate or not vaccinate, and the payoff of each state is some function that roughly capture the average preferences or utilities of all the individuals in the population of that state. We assume that the payoff function is implicit in the behavioral data and that we can learn it or infer it by trying to "rationalize" each state's behavior in the given dataset of vaccination rates. Said differently, here, we do not know the exact game the agents are playing and hence we do not know their payoffs. However, we have data that potentially specify the mixed-strategies of the agents. Some may correspond to an (equilibrium) outcome of the game. Therefore, using the vaccination rates as mixed-strategies, we can learn a game that could partially capture some of these mixed-strategies as outcomes, or, as we view it, *mixed-strategy Nash equilibria (MSNE)* of the game. In addition, we do expect some of these mixed-strategies to be noisy. Therefore, we assume that some of these mixed-strategies may be approximate or ϵ-MSNE of the games and try to find ϵ as small as possible to capture the variations.

In this work, not only we introduce a general machine learning (generative) framework to learn any arbitrary game given the data, we also provide a way to learn a class of *generalized interdependent security (α-IDS) games.* The α-IDS games are introduced in Chapter 3 and we will be using α-IDS games to model vaccination decisions of the players.

We now describe the way we use the CDC data, and delay the details about our learning framework and the mechanisms we derive to learn α-IDS games. Using the 2009-2010 states H1N1 vaccination percentages and their standard deviations, we generate (up to) 5,000 (mixed-strategy) examples according to normal distributions with means and standard deviations of the states. Each example represents a single mixed-strategy profile of the 48-state in the continental USA (i.e., excluding Alaska and Hawaii). Given these examples, we aim to learn an α-IDS game that would best explain the generated data.

Our main interest for learning games is the ability they provide to potentially interpret what would happen at an MSNE, even when the given data may not be an exact MSNE, or be noisy. The mixed-strategies of the state agents in our data may not correspond to the optimal equilibrium strategies. Therefore, we may want to infer, for example, the (real) behavior at the level of states at (either exact or near) equilibrium from noisy data, in which not all examples may belong to the set of approximate MSNE of some game.

Thus, given the learned games, we can run a version of some learning-heuristics/regret-minimization [Fudenberg and Levine, 1998], in which we use the average vaccination rates as the initial mixed-strategy profiles to compute ϵ-MSNE in these games. We expect that as ϵ goes to zero, we would be able to capture the true equilibrium strategies of the state agents.

Contribution. We conclude the introduction with a summary of our contributions. Our interest in this work is learning games from observed mixed-strategy data. In contrast to previous work, in which the data are the actions or pure strategies of the players [Honorio and Ortiz, 2014], we are dealing with data that summarize the actions of all the individuals within a state's population using rates. In our model, we view each rate as representing the mixed strategy of each state agent.

In game-theoretic terms, we view these probabilities collectively as (approximate) MSNE. In particular, we

- propose and introduce a machine learning (generative) framework to learn a game given the data;
- show that, under some mild conditions, maximizing the log-likelihood of the game is equivalent to maximizing the number of (approximate) MSNE under our framework;
- use our framework to derive a heuristic to learn α-IDS games given the CDC vaccination data; and

- experimentally show that our framework and learning heuristic are effective for inferring α-IDS games, and may be able to provide insight into the behavior of state agents.

Related Work The closest work to ours is those of Honorio and Ortiz [2014] where they provide a general machine learning framework to learn the structure and parameters of games from discrete (e.g., "Yes/No" responses) behavioral data. Moreover, they demonstrate their framework on learning influence games [Irfan and Ortiz, 2014] using congressional voting data. For the sake of completeness, all other previous methods assume that the actions and payoffs are observable in the data [Wright and Leyton-Brown, 2010, 2012, Gao and Pfeffer, 2010, Vorobeychik et al., 2007, Ficici et al., 2008, Duong et al., 2009, 2012] while others are interested in predicting future behavior from the past behavior (system dynamics) [Kearns and Wortman, 2008, Ziebart et al., 2010]. We refer the reader to the related work section of Honorio and Ortiz [2014] for a more detailed discussion.

6.2 A Framework to Learn Games from Data

Let $V = \{1, 2, ..., n\}$ be a set of players. For each $i \in V$, let A_i be the set of actions/pure-strategies available to i and $u_i : \times_{j \in V} A_j \to \mathbb{R}$ be the payoff of i given the actions of i and other $V - \{i\}$ agents. Let X_i be the set of mixed-strategies of i, which is a simplex over A_i, and denote by $u_i(x) \equiv \mathbf{E}_{a \sim x}[u_i(x)]$ the expectation over the probabilities x_i of playing the pure-actions. Recall from Chapter 1.2 that a mixed-strategy profile $x^* \in \times_{i=1}^{n} X_i$ is a *mixed-strategy Nash equilibrium (MSNE)* of a non-cooperative game [von Neumann and Morgenstern, 1944, Nash, 1950, 1951] if, for each player i, $x_i^* \in \arg\max_{x_i \in X_i} u_i(x_i, x_{-i}^*)$, where $x_{-i}^* \equiv (x_1^*, x_2^*, \ldots, x_{i-1}^*, x_{i+1}^*, \ldots, x_n^*)$. We denote the set of all MSNE of a game $\mathcal{G}$ as

$$\mathcal{NE}(\mathcal{G}) \equiv \{x^* \mid \forall i,\ x_i^* \in \arg\max_{x_i \in X_i} u_i(x_i, x_{-i}^*)\}$$

For the following definition, we assume that the utilities are normalized between 0 and 1. Given $\epsilon > 0$, a mixed-strategy profile $x^* \in X = \times_{i=1}^{n} X_i$ is an ϵ-MSNE of a non-cooperative game if, for each player i, $x_i^* \in$

arg $\max_{x_i \in X_i} u_i(x_i, x^*_{-i}) - \epsilon$. We denote the set of all ϵ-MSNE of a game $\mathcal{G}$ as

$$\begin{aligned}\mathcal{NE}^{\epsilon}(\mathcal{G}) \equiv \{x^* \mid \forall i,\ & u_i(x^*_i, x^*_{-i}) \geq u_i(0, x^*_{-i}) - \epsilon \text{ and} \\ & u_i(x^*_i, x^*_{-i}) \geq u_i(1, x^*_{-i}) - \epsilon\}\,.\end{aligned}$$

For the rest of the chapter, we assume that each agent has two actions and the action set of each of the agents is either 0 or 1 (i.e., $A_i = \{0, 1\}$ for all i). As such, the mixed-strategies of the agents are in $[0, 1]$ (i.e., $X_i = [0, 1]$ and with probability $x_i \in X_i$, player i plays action 1).

It is easy to see that $\mathcal{NE}(\mathcal{G}) \subseteq \mathcal{NE}^{\epsilon}(\mathcal{G}) \subseteq \mathcal{NE}^{\epsilon'}(\mathcal{G})$ for all $0 < \epsilon < \epsilon'$. Note that an ϵ-MSNE might not be close to any exact MSNE.

6.2.1 A Generative Model for Behavioral Data on Joint-mixed-strategies

We adopt a similar learning approach to that of Honorio and Ortiz [2014]. However, this time the generative model of behavioral data is over the set of mixed-strategy profile. Hence, a *probability density function (PDF)* over the n-dimensional hypercube $[0, 1]^n$ now defines the generative model. We point out that our results are extensions analogous, but non-trivial, to those of Honorio and Ortiz [2014] in a continuous space, as oppose to the use of a *probability mass function (PMF)* over the set of pure-strategy profiles $\{0, 1\}^n$, a discrete space.

We begin by discussing the measurability of $\mathcal{NE}^{\epsilon}(\mathcal{G})$ of an arbitrary game $\mathcal{G}$, which is an important component of our generative model.

Measurability of $\mathcal{NE}^{\epsilon}(\mathcal{G})$

Let (M, d) be a metric space where $M = [0, 1]^n \subseteq \mathbb{R}^n$ and $d : M \times M \to \mathbb{R}$ is any well-defined distance function on M such that:

1. (Non-Negativity) For each $x, y \in M$, $d(x, y) \geq 0$, and $d(x, y) = 0$ if $x = y$,

2. (Symmetry) For each $x, y \in M$, $d(x, y) = d(y, x)$,

3. (Triangle-Inequaility) For each $x, y, z \in M$, $d(x, z) \leq d(x, y) + d(y, z)$.

The Borel σ-algebra $\mathcal{B} = \mathcal{B}(M)$ is the smallest σ-algebra in M that contains all open subsets of M.

Recall that an open subset $U \subset M$ of a metric space (M, d) *open set* with respect to the metric space (M, d) if, for every $p \in U$, there is some $\delta > 0$ such that

$$\mathcal{B}_\delta(p) \subset U,$$

where

$$\mathcal{B}_\delta(p) = \{q \in M : d(p, q) < \delta\}.$$

Moreover, the σ-algebra $\mathcal{B}$ is a nonempty collection of subsets of M such that

1. M is in $\mathcal{B}$;
2. (Closed under complement) If $A \in \mathcal{B}$, the complement of A is in $\mathcal{B}$; that is $X \setminus A \in \mathcal{B}$;
3. (Closed under countable unions) If $A_1, A_2, ...A_n$ are in $\mathcal{B}$ for some n, then $A = A_1 \cup A_2 \cup ... \cup A_n$ is in $\mathcal{B}$.

The Borel σ-algebra $\mathcal{B}$ is the smallest σ-algebra in M that contains all open subsets of M. The elements of $\mathcal{B}$ are called the Borel sets of M. Therefore, a Borel measure is any measure $\mu : \mathcal{B} \to \mathbb{R}$ that maps the Borel sets to some real number.

Let $\mathcal{G}$ be a game and the approximation parameter $\epsilon \geq 0$. We want to show that $\mathcal{NE}^\epsilon(\mathcal{G})$ is (Borel) μ-measurable.

Lemma 17. *The set of ϵ-MSNE, $\mathcal{NE}^\epsilon(\mathcal{G})$, is (Borel) μ-measurable for any game $\mathcal{G}$ and any $\epsilon \geq 0$. That is, the open set(s) of $\mathcal{NE}^\epsilon(\mathcal{G})$ is (are) in $\mathcal{B}$.*

Proof. Recall that $\mathcal{NE}^\epsilon(\mathcal{G})$

$$\begin{aligned} &\equiv \{x^* \mid \forall i,\ u_i(x_i^*, x_{-i}^*) \geq u_i(0, x_{-i}^*) - \epsilon \text{ and } u_i(x_i^*, x_{-i}^*) \geq u_i(1, x_{-i}^*) - \epsilon\} \\ &\equiv \mathcal{BR}_1^\epsilon(\mathcal{G}) \cap \mathcal{BR}_2^\epsilon(\mathcal{G}) \cap \cdots \cap \mathcal{BR}_n^\epsilon(\mathcal{G}), \end{aligned}$$

where

$$\mathcal{BR}_i^\epsilon(\mathcal{G}) = \{x \mid u_i(x_i, x_{-i}) \geq u_i(0, x_{-i}) - \epsilon \text{ and } u_i(x_i, x_{-i}) \geq u_i(1, x_{-i}) - \epsilon\}$$

for $i \in V$. Notice that $\mathcal{BR}_i^\epsilon(\mathcal{G})$ of each player i is formed by two linear inequalities and $\mathcal{BR}_i^\epsilon(\mathcal{G}) \subseteq [0, 1]^n$ which is closed and bounded. The intersection of closed and bounded regions is still closed and bounded. It follows that $\mathcal{NE}^\epsilon(\mathcal{G}) \subseteq [0,1]^n$. It is clear that the open set(s) of $\mathcal{NE}^\epsilon(\mathcal{G})$ is (are) in $\mathcal{B}$. Notice that $\mathcal{NE}^\epsilon(\mathcal{G})$ might contain multiple closed and bounded components but the open set of each individual component is in $\mathcal{B}$. Therefore, $\mathcal{NE}^\epsilon(\mathcal{G})$ is measurable. □

In the following, we denote the μ-measure of $\mathcal{NE}^\epsilon(\mathcal{G})$ by $|\mathcal{NE}^\epsilon(\mathcal{G})| \equiv \mu(\mathcal{NE}(\mathcal{G}))$. We assume the statistical process generating the data is a simple mixture model: i.e., with probability $q \in (0,1)$, the process generates/outputs a mixed-strategy profile x by drawing uniformly at random from the set $\mathcal{NE}^\epsilon(\mathcal{G})$; with probability $1-q$, the process generates a mixed-strategy profile x by drawing uniformly at random from $\overline{\mathcal{NE}^\epsilon(\mathcal{G})} \equiv [0,1]^n - \mathcal{NE}^\epsilon(\mathcal{G})$, the complement of the $\mathcal{NE}^\epsilon(\mathcal{G})$. Said differently, our generative model of behavioral data based on mixed-strategy profile is a mixture model with mixture parameter q and mixture components defined in terms of the approximation parameter $\epsilon > 0$ and a game $\mathcal{G}$. Note that, in our context, because $\mu([0,1]^n) = 1$, we can view the Borel measure μ as a probability measure. From now on, all references to measures are to the Borel (probability) measure, and μ denotes such measure. More formally, the PDF f for the generative model with parameters $(q, \mathcal{G}, \epsilon)$ over the hypercube of joint-mixed-strategies $[0,1]^n$ is

$$f_{(q,\mathcal{G},\epsilon)}(x) \equiv q\frac{\mathbb{1}[x \in \mathcal{NE}^\epsilon(\mathcal{G})]}{|\mathcal{NE}^\epsilon(\mathcal{G})|} + (1-q)\frac{\mathbb{1}[x \notin \mathcal{NE}^\epsilon(\mathcal{G})]}{1 - |\mathcal{NE}^\epsilon(\mathcal{G})|}, \tag{6.1}$$

for all $x \in [0,1]^n$. It is possible that the measure of $\mathcal{NE}^\epsilon(\mathcal{G})$ is zero. As such, for large enough ϵ, the measure is of it will be greater than zero. However, one can formally show that for any $\epsilon > 0$, $|\mathcal{NE}^\epsilon(\mathcal{G})| > 0$. The key to show this fact is to realize that there is at least one MSNE in $\mathcal{NE}^\epsilon(\mathcal{G})$ for any $\epsilon > 0$. Using that MSNE, we can find a region surrounding it and this region is determined by the value of ϵ. We do not present a formal proof here because this is not our main focus. Instead, we concentrate on learning the parameters of games with $|\mathcal{NE}^\epsilon(\mathcal{G})| > 0$ for some $\epsilon > 0$.

In order for Equation 6.1 to be valid, if $\epsilon = 1$ or $\mathcal{NE}^\epsilon(\mathcal{G}) = [0,1]^n$, then we need to require that $q = 1$. Note that $\mathcal{NE}^\epsilon(\mathcal{G}) = \emptyset$ is impossible since every game has at least one MSNE, by Nash's seminal result [Nash, 1950, 1951].

We assume that the behavioral data are i.i.d. instances drawn according to $f_{(q,\mathcal{G},\epsilon)}$.

Definition 16. (Trivial and Non-trivial Games) *We say that a game $\mathcal{G}$ is* trivial *if and only if $|\mathcal{NE}^{\epsilon}(\mathcal{G})| \in \{0, 1\}$ and* non-trivial *if and only if $|\mathcal{NE}^{\epsilon}(\mathcal{G})| \in (0, 1)$.*

Let $\pi^{\epsilon}(\mathcal{G})$ be the *true proportion of ϵ-MSNE* in the game $\mathcal{G}$ where

$$\pi^{\epsilon}(\mathcal{G}) \equiv |\mathcal{NE}^{\epsilon}(\mathcal{G})| \, . \tag{6.2}$$

The following set of propositions is analogous to those Honorio and Ortiz [2014] state and establishes the fact that there are different games with the same set of ϵ-MSNE. Said differently, the game $\mathcal{G}$ is not identifiable with respect to the generative model $f_{(q,\mathcal{G},\epsilon)}$ defined in Equation 6.1. We side-step the non-identifiability of $\mathcal{G}$ with respect to $f_{(q,\mathcal{G},\epsilon)}$ using a common practice in machine learning (ML): we invoke the *Principle of Ockham's Razor* for model (i.e., game) selection. In general, experts in the respective field (e.g., epidemiology) would provide the necessary bias for learning. Here, we impose a particular bias that induces "sparse" or "compactly representable" games, as we define formally in a later section. We note, however, that the games are identifiable in terms of their ϵ-MSNE, with respect to $f_{(q,\mathcal{G},\epsilon)}$, which is our main interest, as we show and discuss later.

Proposition 11. *Given the approximation parameter $\epsilon > 0$ and a non-trivial game $\mathcal{G}$, the mixture parameter $q > \pi^{\epsilon}(\mathcal{G})$ if and only if $f_{(q,\mathcal{G},\epsilon)}(x_1) > f_{(q,\mathcal{G},\epsilon)}(x_2)$ for any $x_1 \in \mathcal{NE}^{\epsilon}(\mathcal{G})$ and $x_2 \notin \mathcal{NE}^{\epsilon}(\mathcal{G})$.*

Proof. Suppose that $q > \pi^{\epsilon}(\mathcal{G})$. For any $x_1 \in \mathcal{NE}^{\epsilon}(\mathcal{G})$, $f_{(q,\mathcal{G},\epsilon)}(x_1) = \frac{q}{|\mathcal{NE}^{\epsilon}(\mathcal{G})|}$. On the other hand, for any $x_2 \notin \mathcal{NE}^{\epsilon}(\mathcal{G})$, $f_{(q,\mathcal{G},\epsilon)}(x_2) = \frac{1-q}{1-|\mathcal{NE}^{\epsilon}(\mathcal{G})|}$. Since $q > \pi^{\epsilon}(\mathcal{G})$, we have

$$f_{(q,\mathcal{G},\epsilon)}(x_1) = \frac{q}{|\mathcal{NE}^{\epsilon}(\mathcal{G})|} > 1 > \frac{1-q}{1-|\mathcal{NE}^{\epsilon}(\mathcal{G})|} = f_{(q,\mathcal{G},\epsilon)}(x_2),$$

where the second inequality is because $1 - q < 1 - |\mathcal{NE}^{\epsilon}(\mathcal{G})|$.

Now we suppose that for any $x_1 \in \mathcal{NE}^{\epsilon}(\mathcal{G})$ and $x_2 \notin \mathcal{NE}^{\epsilon}(\mathcal{G})$, $f_{(q,\mathcal{G},\epsilon)}(x_1) > f_{(q,\mathcal{G},\epsilon)}(x_2)$. It follows that

$$f_{(q,\mathcal{G},\epsilon)}(x_1) = \frac{q}{|\mathcal{NE}^{\epsilon}(\mathcal{G})|} > \frac{1-q}{1-|\mathcal{NE}^{\epsilon}(\mathcal{G})|} = f_{(q,\mathcal{G},\epsilon)}(x_2).$$

The above inequality holds if we have

$$\begin{aligned}
&\frac{q}{|\mathcal{NE}^{\epsilon}(\mathcal{G})|} > \frac{1-q}{1-|\mathcal{NE}^{\epsilon}(\mathcal{G})|} \\
\iff & q(1-|\mathcal{NE}^{\epsilon}(\mathcal{G})|) > (1-q)|\mathcal{NE}^{\epsilon}(\mathcal{G})| \\
\iff & q - q|\mathcal{NE}^{\epsilon}(\mathcal{G})| > |\mathcal{NE}^{\epsilon}(\mathcal{G})| - q|\mathcal{NE}^{\epsilon}(\mathcal{G})| \\
\iff & q > |\mathcal{NE}^{\epsilon}(\mathcal{G})|
\end{aligned}$$

This concludes the proof. □

Definition 17. *Given the approximation parameter $\epsilon > 0$, we say the games $\mathcal{G}_1$ and $\mathcal{G}_2$ are* ϵ-approximation-equivalent, *or* ϵ-equivalent, *if and only if their approximate Nash equilibrium sets are identical, i.e.: $\mathcal{G}_1 \equiv_{\epsilon-\text{MSNE}} \mathcal{G}_2 \Leftrightarrow \mathcal{NE}^{\epsilon}(\mathcal{G}_1) = \mathcal{NE}^{\epsilon}(\mathcal{G}_2)$.*

Lemma 18. *Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two non-trivial games. Given the approximation parameter $\epsilon > 0$, for some mixture parameter $q > \max(\pi^{\epsilon}(\mathcal{G}_1), \pi^{\epsilon}(\mathcal{G}_2))$, $\mathcal{G}_1$ and $\mathcal{G}_2$ are ϵ-equivalent if and only if they induce the same PDF over the mixed-strategies of the agents, which belong to $[0,1]^n$ by definition. Moreover, $\mathcal{G}_1 \equiv_{\epsilon-\text{MSNE}} \mathcal{G}_2 \Leftrightarrow \forall x \ f_{(q,\mathcal{G}_1,\epsilon)}(x) = f_{(q,\mathcal{G}_2,\epsilon)}(x)$.*

Proof. Suppose that $\mathcal{G}_1 \equiv_{\epsilon-MSNE} \mathcal{G}_2$. It follows that $\mathcal{NE}^{\epsilon}(\mathcal{G}_1) = \mathcal{NE}^{\epsilon}(\mathcal{G}_2)$. By the PDF defined in Equation 6.1, for all $x \in \mathcal{NE}^{\epsilon}(\mathcal{G}_1)$,

$$f_{(q,\mathcal{G}_1,\epsilon)}(x) = \frac{q}{|\mathcal{NE}^{\epsilon}(\mathcal{G}_1)|} = \frac{q}{|\mathcal{NE}^{\epsilon}(\mathcal{G}_2)|} = f_{(q,\mathcal{G}_2,\epsilon)}(x),$$

and for all $x \notin \mathcal{NE}^{\epsilon}(\mathcal{G}_1)$,

$$f_{(q,\mathcal{G}_1,\epsilon)}(x) = \frac{1-q}{1-|\mathcal{NE}^{\epsilon}(\mathcal{G}_1)|} = \frac{1-q}{1-|\mathcal{NE}^{\epsilon}(\mathcal{G}_2)|} = f_{(q,\mathcal{G}_2,\epsilon)}(x).$$

Therefore, for all x, $f_{(\mathcal{G}_1,q,\epsilon)}(x) = f_{(\mathcal{G}_2,q,\epsilon)}(x)$.

Now suppose that $\forall x \ f_{(\mathcal{G}_1,q,\epsilon)}(x) = f_{(\mathcal{G}_2,q,\epsilon)}(x)$. For the sake of contradiction, assume there is some y such that $y \in \mathcal{G}_1$ and $y \notin \mathcal{G}_2$. Since $f_{(\mathcal{G}_1,q,\epsilon)}(y) = f_{(\mathcal{G}_2,q,\epsilon)}(y)$, we have that

$$\begin{aligned}
&\frac{q}{|\mathcal{NE}^{\epsilon}(\mathcal{G}_1)|} = \frac{1-q}{1-|\mathcal{NE}^{\epsilon}(\mathcal{G}_2)|} \\
\iff & q(1-|\mathcal{NE}^{\epsilon}(\mathcal{G}_2)|) = (1-q)|\mathcal{NE}^{\epsilon}(\mathcal{G}_1)| \\
\iff & q(1-|\mathcal{NE}^{\epsilon}(\mathcal{G}_2)|) + q|\mathcal{NE}^{\epsilon}(\mathcal{G}_1)| = |\mathcal{NE}^{\epsilon}(\mathcal{G}_1)| \\
\iff & q = \frac{|\mathcal{NE}^{\epsilon}(\mathcal{G}_1)|}{1+|\mathcal{NE}^{\epsilon}(\mathcal{G}_1)| - |\mathcal{NE}^{\epsilon}(\mathcal{G}_2)|}.
\end{aligned}$$

Because $q > \max(\pi^\epsilon(\mathcal{G}_1), \pi^\epsilon(\mathcal{G}_2))$, we have that $q > \max\left(|\mathcal{NE}^\epsilon(\mathcal{G}_1)|, |\mathcal{NE}^\epsilon(\mathcal{G}_2)|\right)$. Moreover,

$$\frac{|\mathcal{NE}^\epsilon(\mathcal{G}_1)|}{1 + |NE^\epsilon(\mathcal{G}_1)| - |\mathcal{NE}^\epsilon(\mathcal{G}_2)|} > \max(|\mathcal{NE}^\epsilon(\mathcal{G}_1)|, |\mathcal{NE}^\epsilon(\mathcal{G}_2)|).$$

If $\max(|\mathcal{NE}^\epsilon(\mathcal{G}_1)|, |\mathcal{NE}^\epsilon(\mathcal{G}_2)|) = |\mathcal{NE}^\epsilon(\mathcal{G}_1)|$, then

$$\begin{aligned} &\frac{|\mathcal{NE}^\epsilon(\mathcal{G}_1)|}{1 + |NE^\epsilon(\mathcal{G}_1)| - |\mathcal{NE}^\epsilon(\mathcal{G}_2)|} > |\mathcal{NE}^\epsilon(\mathcal{G}_1)| \\ \iff &|\mathcal{NE}^\epsilon(\mathcal{G}_1)| > \left(1 + |NE^\epsilon(\mathcal{G}_1)| - |\mathcal{NE}^\epsilon(\mathcal{G}_2)|\right) |\mathcal{NE}^\epsilon(\mathcal{G}_1)|, \end{aligned}$$

which is a contradiction because $|NE^\epsilon(\mathcal{G}_1)| - |\mathcal{NE}^\epsilon(\mathcal{G}_2)| \geq 0$.

On the other hand, if $\max(|\mathcal{NE}^\epsilon(\mathcal{G}_1)|, |\mathcal{NE}^\epsilon(\mathcal{G}_2)|) = |\mathcal{NE}^\epsilon(\mathcal{G}_2)|$, then

$$\begin{aligned} &\frac{|\mathcal{NE}^\epsilon(\mathcal{G}_1)|}{1 + |NE^\epsilon(\mathcal{G}_1)| - |\mathcal{NE}^\epsilon(\mathcal{G}_2)|} > |\mathcal{NE}^\epsilon(\mathcal{G}_2)| \\ \iff &|\mathcal{NE}^\epsilon(\mathcal{G}_1)| > \left(1 + |NE^\epsilon(\mathcal{G}_1)| - |\mathcal{NE}^\epsilon(\mathcal{G}_2)|\right) |\mathcal{NE}^\epsilon(\mathcal{G}_2)| \\ \iff &|\mathcal{NE}^\epsilon(\mathcal{G}_1)| > |\mathcal{NE}^\epsilon(\mathcal{G}_2)| + |NE^\epsilon(\mathcal{G}_1)||\mathcal{NE}^\epsilon(\mathcal{G}_2)| - |\mathcal{NE}^\epsilon(\mathcal{G}_2)|^2 \\ \iff &0 > |\mathcal{NE}^\epsilon(\mathcal{G}_2)| + |NE^\epsilon(\mathcal{G}_1)||\mathcal{NE}^\epsilon(\mathcal{G}_2)| - |\mathcal{NE}^\epsilon(\mathcal{G}_2)|^2 - |\mathcal{NE}^\epsilon(\mathcal{G}_1)| \\ \iff &0 > (|\mathcal{NE}^\epsilon(\mathcal{G}_2)| - |\mathcal{NE}^\epsilon(\mathcal{G}_1)|) - |\mathcal{NE}^\epsilon(\mathcal{G}_2)|(|\mathcal{NE}^\epsilon(\mathcal{G}_2)| - |NE^\epsilon(\mathcal{G}_1)|) \\ \iff &0 > (1 - |\mathcal{NE}^\epsilon(\mathcal{G}_2)|)(|\mathcal{NE}^\epsilon(\mathcal{G}_2)| - |\mathcal{NE}^\epsilon(\mathcal{G}_1)|), \end{aligned}$$

which is a contradiction because $|\mathcal{NE}^\epsilon(\mathcal{G}_2)| > 0$ and $|\mathcal{NE}^\epsilon(\mathcal{G}_2)| - |\mathcal{NE}^\epsilon(\mathcal{G}_1)| > 0$. Therefore such y does not exists and $\mathcal{G}_1 \equiv_{\epsilon-MSNE} \mathcal{G}_2$. □

6.2.2 Learning Parameters of the Games via MLE

In this section, we present a way to estimate the parameters of a graphical game from data. We assume the data are i.i.d. draws from the generative model just defined above.

For the following, we recall the Kullback-Leibler (KL) divergence between two Bernoulli distributions with parameters $p_1, p_2 \in (0, 1)$, which, using common practice to simplify the presentation, we denote by

$$\mathrm{KL}(p_1 \| p_2) \equiv p_1 \log \frac{p_1}{p_2} + (1 - p_1) \log \frac{1 - p_1}{1 - p_2} \; .$$

Given a dataset $D = \{x^{(1)}, ..., x^{(m)}\}$ drawn *i.i.d.* according to $f_{q,\mathcal{G},\epsilon}$, let $\widehat{\pi}^{\epsilon}(\mathcal{G})$ be the *empirical proportion of ϵ-MSNE*, i.e.,

$$\widehat{\pi}^{\epsilon}(\mathcal{G}) \equiv \frac{1}{m} \sum_{l=1}^{m} \mathbb{1}\big[x^{(l)} \in \mathcal{NE}^{\epsilon}(\mathcal{G})\big]$$

Proposition 12. (Maximum-likelihood Estimation) *The tuple $(\widehat{\mathcal{G}}, \widehat{q}, \widehat{\epsilon})$ is a* maximum likelihood estimator (MLE)*, with respect to dataset D, for the parameters of the generative model $f_{(q,\mathcal{G},\epsilon)}$, as defined in Equation 6.1 if and only if $\widehat{q} = \min\left(\widehat{\pi}^{\widehat{\epsilon}}(\widehat{\mathcal{G}}), 1 - \frac{1}{2m}\right)$, and $(\widehat{\mathcal{G}}, \widehat{\epsilon}) \in \arg\max_{(\mathcal{G},\epsilon)} \mathrm{KL}(\widehat{\pi}^{\epsilon}(\mathcal{G}) \| \pi^{\epsilon}(\mathcal{G}))$.*

Proof. For simplicity, we denote $\mathcal{NE}^{\epsilon} = \mathcal{NE}^{\epsilon}(\mathcal{G})$, $\pi^{\epsilon} \equiv \pi^{\epsilon}(\mathcal{G})$, and $\hat{\pi}^{\epsilon} \equiv \hat{\pi}^{\epsilon}(\mathcal{G})$. For a nontrivial $\mathcal{G}$, $\log f_{(\mathcal{G},q,\epsilon)}(x^{(l)}) = \log \frac{q}{|\mathcal{NE}^{\epsilon}|}$ for $x^{(l)} \in \mathcal{NE}^{\epsilon}$ and $\log f_{(\mathcal{G},q,\epsilon)}(x^{(l)}) = \log \frac{1-q}{1-|\mathcal{NE}^{\epsilon}|}$ for $x^{(l)} \notin \mathcal{NE}^{\epsilon}$. The average log-likelihood

$$\begin{aligned} \hat{\mathbb{L}}(\mathcal{G}, q, \epsilon) &= \frac{1}{m} \sum_{l=1}^{m} \log f_{(\mathcal{G},q,\epsilon)}(x^{(l)}) \\ &= \hat{\pi}^{\epsilon} \log \frac{q}{|\mathcal{NE}^{\epsilon}|} + (1 - \hat{\pi}^{\epsilon}) \log \frac{1-q}{1-|\mathcal{NE}^{\epsilon}|} \\ &= \hat{\pi}^{\epsilon} \log \frac{q}{\pi^{\epsilon}} + (1 - \hat{\pi}^{\epsilon}) \log \frac{1-q}{1-\pi^{\epsilon}}. \end{aligned}$$

By adding and subtracting the term $\hat{\pi}^{\epsilon} \log \hat{\pi}^{\epsilon}$, we have

$$\begin{aligned} \hat{\mathbb{L}}(\mathcal{G}, q, \epsilon) &= \hat{\pi}^{\epsilon} \log \frac{q}{\pi^{\epsilon}} + (1 - \hat{\pi}^{\epsilon}) \log \frac{1-q}{1-\pi^{\epsilon}} + \hat{\pi}^{\epsilon} \log \hat{\pi}^{\epsilon} - \hat{\pi}^{\epsilon} \log \hat{\pi}^{\epsilon} \\ &= \hat{\pi}^{\epsilon} \log q - \hat{\pi}^{\epsilon} \log \pi^{\epsilon} + (1 - \hat{\pi}^{\epsilon}) \log \frac{1-q}{1-\pi^{\epsilon}} + \hat{\pi}^{\epsilon} \log \hat{\pi}^{\epsilon} - \hat{\pi}^{\epsilon} \log \hat{\pi}^{\epsilon} \\ &= \hat{\pi}^{\epsilon} \log \frac{\hat{\pi}^{\epsilon}}{\pi^{\epsilon}} - \hat{\pi}^{\epsilon} \log \frac{\hat{\pi}^{\epsilon}}{q} + (1 - \pi^{\epsilon}) \log \frac{1-q}{1-\pi^{\epsilon}}. \end{aligned}$$

Similarly, we add and subtract the term $(1 - \hat{\pi}^{\epsilon}) \log(1 - \hat{\pi}^{\epsilon})$, and we have

$$\begin{aligned} \hat{\mathbb{L}}(\mathcal{G}, q, \epsilon) &= \hat{\pi}^{\epsilon} \log \frac{\hat{\pi}^{\epsilon}}{\pi^{\epsilon}} + (1 - \hat{\pi}^{\epsilon}) \log \frac{1-\hat{\pi}^{\epsilon}}{1-\pi^{\epsilon}} - \hat{\pi}^{\epsilon} \log \frac{\hat{\pi}^{\epsilon}}{q} - (1 - \hat{\pi}^{\epsilon}) \log \frac{1-\hat{\pi}^{\epsilon}}{1-q} \\ &= \mathrm{KL}(\hat{\pi}^{\epsilon} \| \pi^{\epsilon}) - \mathrm{KL}(\hat{\pi}^{\epsilon} \| q). \end{aligned}$$

To maximize the log-likelihood, the term $KL(\hat{\pi}^{\epsilon} || \hat{q}) = 0$ if and only if $\hat{q} = \hat{\pi}^{\epsilon}$. If $\hat{\pi}^{\epsilon} = 1$, then we shrink $\hat{q} = 1 - \frac{1}{2m}$ to make sure the PDF is valid. Therefore, we find that the MLE is a tuple $(\widehat{\mathcal{G}}, \widehat{\epsilon})$, where $\widehat{\mathcal{G}} \in \arg\max_{\mathcal{G}} \mathrm{KL}(\hat{\pi}^{\hat{\epsilon}}(\mathcal{G}) \| \pi^{\hat{\epsilon}}(\mathcal{G}))$. □

Let us make a few observations that follow immediately from the MLE expression given above. First, if $\epsilon \geq 1$, then $\pi^\epsilon(\mathcal{G}) = 1$ for all games $\mathcal{G}$, which implies then $\widehat{\pi}^\epsilon(\mathcal{G}) = 1$ for all games $\mathcal{G}$. Hence, if $\widehat{\epsilon} \geq 1$ the resulting KL value is zero, so that $\widehat{\mathcal{G}}$ could be *any* game. Similarly, if $\pi^{\widehat{\epsilon}}(\widehat{\mathcal{G}}) = 0$ then we have $\widehat{\pi}^{\widehat{\epsilon}}(\widehat{\mathcal{G}}) = 0$, so that once again the resulting KL value is zero. Hence, $\widehat{\mathcal{G}}$ could be *any* game. Said differently, in summary, if *any* trivial game is an MLE, then *every* game, trivial or non-trivial, is also an MLE. Therefore, we can always find non-trivial games corresponding to some MLE: the set of MLEs always contain a tuple corresponding to a non-trivial game.

An informal interpretation of the MLE problem is that, assuming we can keep the true proportion of ϵ-MSNE low, the learning problems becomes one of trying to infer a game that captures as much of the mixed-strategy examples in the dataset as ϵ-MSNE, but without implicitly adding more ϵ-MSNE than it needs to. Thus, formulating the learning problem using MLE brings out the fundamental tradeoff in ML between model complexity and generalization ability (or "goodness-of-fit"), despite the simplicity of our generative model.

One problem with the exact KL-based formulation of the MLE presented above is that dealing with $\pi^\epsilon(\mathcal{G})$, even if it is well-defined (i.e., the set $\mathcal{NE}^\epsilon(\mathcal{G})$ has positive measure). The following lemma provides bounds on the KL divergence that will prove useful in our setting.

Lemma 19. *Given a non-trivial game $\mathcal{G}$ with $0 < \pi^\epsilon(\mathcal{G}) < \widehat{\pi}^\epsilon(\mathcal{G})$, we can upper and lower bound the KL divergence as*

$$-\widehat{\pi}^\epsilon(\mathcal{G}) \log \pi^\epsilon(\mathcal{G}) - \log 2 < \mathrm{KL}(\widehat{\pi}^\epsilon(\mathcal{G}) \| \pi^\epsilon(\mathcal{G})) < -\widehat{\pi}^\epsilon(\mathcal{G}) \log \pi^\epsilon(\mathcal{G}) \,.$$

Proof. The proof for the above lemma follows closely of the same argument of a similar bound in Honorio and Ortiz [2014]. The main distinction is that we are dealing with ϵ-MSNE. We reproduce it here for completeness.

For simplicity, we denote $\pi^\epsilon \equiv \pi^\epsilon(\mathcal{G})$ and $\hat{\pi}^\epsilon \equiv \hat{\pi}^\epsilon(\mathcal{G})$. From the definition of KL, we have

$$\mathrm{KL}(\hat{\pi}^\epsilon \| \pi^\epsilon) = \hat{\pi}^\epsilon \log \frac{\hat{\pi}^\epsilon}{\pi^\epsilon} + (1 - \hat{\pi}^\epsilon) \log \frac{1 - \hat{\pi}^\epsilon}{1 - \pi^\epsilon}.$$

Since $0 < \pi^\epsilon(\mathcal{G}) < \widehat{\pi}^\epsilon(\mathcal{G})$, we have

$$\alpha(\pi^\epsilon) \equiv \lim_{\hat{\pi}^\epsilon \to 0} \mathrm{KL}(\hat{\pi}^\epsilon \| \pi^\epsilon) = 0 \quad \text{and} \quad \beta(\pi^\epsilon) \equiv \lim_{\hat{\pi}^\epsilon \to 1} \mathrm{KL}(\hat{\pi}^\epsilon \| \pi^\epsilon) = -\log \pi^\epsilon.$$

Moreover, the KL function is convex and

$$\mathrm{KL}(\hat{\pi}^{\epsilon} \| \pi^{\epsilon}) \leq \alpha(\pi^{\epsilon}) + (\beta(\pi^{\epsilon}) - \alpha(\pi^{\epsilon}))\hat{\pi}^{\epsilon} = -\hat{\pi}^{\epsilon} \log \pi^{\epsilon}.$$

Thus, we have our upper bound.

To find a lower bound, we first find $\overline{\hat{\pi}^{\epsilon}}$ such that $\frac{\partial \mathrm{KL}(\hat{\pi}^{\epsilon} \| \pi^{\epsilon})}{\partial \hat{\pi}^{\epsilon}} = \beta(\pi^{\epsilon}) - \alpha(\pi^{\epsilon}) = -\log \pi^{\epsilon}$. Following the derivation, we have

$$\begin{aligned}
\frac{\partial \mathrm{KL}(\hat{\pi}^{\epsilon} \| \pi^{\epsilon})}{\partial \hat{\pi}^{\epsilon}} &= \frac{\partial \left(\hat{\pi}^{\epsilon} \log \frac{\hat{\pi}^{\epsilon}}{\pi^{\epsilon}} + (1 - \hat{\pi}^{\epsilon}) \log \frac{1 - \hat{\pi}^{\epsilon}}{1 - \pi^{\epsilon}} \right)}{\partial \hat{\pi}^{\epsilon}} \\
&= \log \hat{\pi}^{\epsilon} + 1 - \log \pi^{\epsilon} - \log(1 - \hat{\pi}^{\epsilon}) - 1 + \log(1 - \pi^{\epsilon}) \\
&= \log \hat{\pi}^{\epsilon} - \log \pi^{\epsilon} - \log(1 - \hat{\pi}^{\epsilon}) + \log(1 - \pi^{\epsilon}).
\end{aligned}$$

Given this, we find $\overline{\hat{\pi}^{\epsilon}}$ such that

$$\begin{aligned}
&\frac{\partial \mathrm{KL}(\overline{\hat{\pi}^{\epsilon}} \| \pi^{\epsilon})}{\partial \hat{\pi}^{\epsilon}} = -\log \pi^{\epsilon} \\
\iff &\log \overline{\hat{\pi}^{\epsilon}} - \log \pi^{\epsilon} - \log(1 - \overline{\hat{\pi}^{\epsilon}}) + \log(1 - \pi^{\epsilon}) = -\log \pi^{\epsilon} \\
\iff &\log \overline{\hat{\pi}^{\epsilon}} - \log(1 - \overline{\hat{\pi}^{\epsilon}}) + \log(1 - \pi^{\epsilon}) = 0 \\
\iff &\log \frac{\overline{\hat{\pi}^{\epsilon}}}{1 - \overline{\hat{\pi}^{\epsilon}}} = \log \frac{1}{1 - \pi^{\epsilon}} \\
\iff &\frac{\overline{\hat{\pi}^{\epsilon}}}{1 - \overline{\hat{\pi}^{\epsilon}}} = \frac{1}{1 - \pi^{\epsilon}} \\
\iff &\overline{\hat{\pi}^{\epsilon}}(1 - \pi^{\epsilon}) = 1 - \overline{\hat{\pi}^{\epsilon}} \\
\iff &\overline{\hat{\pi}^{\epsilon}} = \frac{1}{2 - \pi^{\epsilon}}
\end{aligned}$$

We now compute the maximum difference between the KL and the upper bound (this will give us a lower bound on the KL). It follows that

$$\begin{aligned}
\lim_{\pi^{\epsilon} \to 0} -\overline{\hat{\pi}^{\epsilon}} \log \pi^{\epsilon} - \mathrm{KL}(\overline{\hat{\pi}^{\epsilon}} \| \pi^{\epsilon}) &= \lim_{\pi^{\epsilon} \to 0} -\overline{\hat{\pi}^{\epsilon}} \log \pi^{\epsilon} - \overline{\hat{\pi}^{\epsilon}} \log \frac{\overline{\hat{\pi}^{\epsilon}}}{\pi^{\epsilon}} - (1 - \overline{\hat{\pi}^{\epsilon}}) \log \frac{1 - \overline{\hat{\pi}^{\epsilon}}}{1 - \pi^{\epsilon}} \\
&= \lim_{\pi^{\epsilon} \to 0} -\overline{\hat{\pi}^{\epsilon}} \log \pi^{\epsilon} - \overline{\hat{\pi}^{\epsilon}} \log \overline{\hat{\pi}^{\epsilon}} + \overline{\hat{\pi}^{\epsilon}} \log \pi^{\epsilon} - (1 - \overline{\hat{\pi}^{\epsilon}}) \log \frac{1 - \overline{\hat{\pi}^{\epsilon}}}{1 - \pi^{\epsilon}} \\
&= \lim_{\pi^{\epsilon} \to 0} -\overline{\hat{\pi}^{\epsilon}} \log \overline{\hat{\pi}^{\epsilon}} - (1 - \overline{\hat{\pi}^{\epsilon}}) \log \frac{1 - \overline{\hat{\pi}^{\epsilon}}}{1 - \pi^{\epsilon}} \\
&= -\frac{1}{2} \log \frac{1}{2} - (1 - \frac{1}{2}) \log \frac{1}{2} \\
&= \log 2.
\end{aligned}$$

Thus, our lower bound is the upper bound minus $\log 2$. This shows that we can bound the KL using the upper bound and the lower bound as claimed. □

From the last lemma, it is easy to see that $\pi^{\epsilon}(\mathcal{G})$ is "low enough" then we can obtain an approximation to the MLE by simply maximizing $\widehat{\pi}^{\epsilon}(\mathcal{G})$ only: i.e., $\arg\max_{\mathcal{G}} \mathrm{KL}(\widehat{\pi}^{\epsilon}(\mathcal{G}) \| \pi^{\epsilon}(\mathcal{G})) \approx \arg\max_{\mathcal{G}} \widehat{\pi}^{\epsilon}(\mathcal{G})$. We implicitly enforce the constraint that $\pi^{\epsilon}(\mathcal{G})$ is "low enough" through regularization or some other way that allows us to introduce bias into the model selection, as it is standard in Machine Learning.

Therefore, we aim to develop techniques to maximize the number of ϵ-MSNE in the data while keeping ϵ as small as possible. In what follows, we will apply our learning framework to infer the parameters of generalized Interdependent Security (α-IDS) games using the CDC vaccination data.

6.3 Application: Learning α-IDS Games

Given the CDC vaccination data, we want to learn a game that would explain the behavior of the agents and how the behavior of the agents affect the behavior of other agents (within the same population). In particular, we are interested in understanding the behavior of an "average" individual in each state when facing the question of "What is the probability that a member of my population will transfer a virus/sickness to me if that member is ill?" Therefore, we want to look at games that model such interaction.

As discussed in Chapter 3, generalized Interdependent Security (α-IDS) games are one of the most motivated and well-studied games to model the investment decisions of agents when facing transfer risks from other agents. As Heal and Kunreuther [2005a] discusses, α-IDS games have applicability in fire protection [Kearns and Ortiz, 2004], and, more importantly in vaccination settings [Heal and Kunreuther, 2005b]. We refer the reader to a recent survey by Laszka et al. [2014] for a broader concept of interdependent security,

In the vaccination setting, each agent decides whether or not to get vaccinated given (1) the agent's implicit and explicit cost of vaccination and loss of getting sick, (2) the vaccination decisions of other agents, and (3) the potential transfer probabilities/risks from other agents. The CDC vaccination data captures the "average" behavior of the people in each state through the vaccination rates, but does not explicitly contain the costs or losses of any individual, nor the transfer risk between individuals. Actually, it does not include the “average” costs, losses, or transfer risks even at the level of whole states. In what follows, we put forward an approach to learn such quantities at the state level from the CDC data on vaccination rates.

6.3.1 Generalized Interdependent Security Games

Recall, in the α-IDS games of n agents, each agent i determines whether or not to invest in protection. Therefore, there are two actions i can play, and we denote $a_i = 1$ if i invests and $a_i = 0$ if i does not invest. We let $a = (a_1, ..., a_n)$ to be the joint-action profile of all agents and a_{-S} to be the joint-action profile of all agents that are not in S. There is a cost of investment C_i and loss L_i associated with the bad event occurring, either through a direct or indirect (transferred) contamination. We denote by p_i the probability that agent i will experience the bad event from a direct contamination and by q_{ji} to be the probability that agent i will experience the bad event due to transfer exposure from agent j (i.e., the probability that agent j will transfer the contamination to i). Moreover, the parameter $\alpha_i \in [0, 1]$ specifies the probability that agent i's investment will not protect that agent against transfers of a bad event. Given the parameters, the *cost function of agent* i is

$$\begin{aligned} M_i(a_i, a_{-i}) \equiv\; & a_i[C_i + \alpha_i r_i(a_{-i})L_i] \\ & + (1 - a_i)[p_i + (1 - p_i)r_i(a_{-i})]L_i \end{aligned}$$

where $r_i(a_{-i}) \equiv 1 - s_i(a_{-i})$ and $s_i(a_{-i}) \equiv \prod_{j \neq i}(a_j + (1 - a_j)(1 - q_{ji}))$ are the overall risk and safety functions of agent i.

As mentioned before, we aim to learn all the model parameters from a given set of observed mixed-strategy profiles, which contain hopefully most, but not necessarily all ϵ-MSNE.

Therefore, we need look at the cost function of the agents in terms of mixed-strategies. Roughly speaking, we can do this by letting x_i be the probability that $a_i = 1$ and take the expectation of the above cost function (i.e., replace all a terms by x). Comparing the cost when $a_i = 1$ and $a_i = 0$, we can derive a best-response correspondence for i.

Therefore, the cost function of player i becomes (in mixed-strategies)

$$\begin{aligned} M_i(x_i, x_{-i}) \equiv\; & x_i[C_i + \alpha_i r_i(x_{-i})L_i] \\ & + (1 - x_i)[p_i + (1 - p_i)r_i(x_{-i})]L_i. \end{aligned}$$

By definition, a mixed-strategy is an ϵ-MSNE in an α-IDS game if and only if

$$M_i(x_i, x_{-i}) - \epsilon \leq M_i(0, x_{-i}) \tag{6.3}$$

$$M_i(x_i, x_{-i}) - \epsilon \leq M_i(1, x_{-i}) \tag{6.4}$$

It follows that from Equation 6.3 and Equation 6.4 that

$$\begin{aligned} x_i[C_i + \alpha_i r_i(x_{\mathrm{Pa}(i)})L_i - (p_i + (1-p_i)r_i(x_{\mathrm{Pa}(i)}))L_i] &\leq \epsilon \\ -(1-x_i)[C_i + \alpha_i r_i(x_{\mathrm{Pa}(i)})L_i - (p_i + (1-p_i)r_i(x_{\mathrm{Pa}(i)}))L_i] &\leq \epsilon \end{aligned}$$

For simplicity, we let $\Delta_i \equiv C_i + \alpha_i r_i(x_{\mathrm{Pa}(i)})L_i - (p_i + (1-p_i)r_i(x_{\mathrm{Pa}(i)}))L_i$.

6.3.2 Learning the Structure and Parameters of α-IDS Games

As we argue in a previous section, we can approximate our MLE objective by maximizing the number of ϵ-MSNE in the data, or equivalently, maximizing $\widehat{\pi}^\epsilon(\mathcal{G})$ over ϵ and $\mathcal{G}$ when the true proportion of the ϵ-MSNE of the game is less than the empirical proportion of the ϵ-MSNE of the dataset (i.e., $0 < \pi^\epsilon(\mathcal{G}) < \widehat{\pi}^\epsilon(\mathcal{G})$). *Below, we empirically show that the true proportion of ϵ-MSNE in α-IDS games is very small.* This would justify Lemma 19 and our method of finding an α-IDS game that maximizes the number of ϵ-MSNE in the dataset.

Figure 6.1 shows the sampled proportional of randomly generated 48-player α-IDS games in various graph structures. In particular, we consider two basic graph structures that specify the transfer risks among the players. The first graph structure results from the geo-spatial adjacency of all states in the U.S.A continental (i.e., excluding Alaska and Hawaii), where each of the 48 players corresponds to a state of the US and the potential transfer risks occur from neighboring states/players. The second graph structure is based on the random graph generation of Erdös and Rényi [1959]. We refer to the latter type of graphs as *ER* graphs. To generate an *ER* graph, we need to specify the number of nodes and a probability $p \in [0, 1]$ that denotes the probability that the drawn ER graph will have an edge between any two nodes. Clearly, a higher p value corresponds to a higher density of the graph. In our case, we use *ER* graph as a way to generate different structures among the 48 players with $p \in \{0.1, 0.2, ..., 0.9\}$. Given the graphs, we generate the

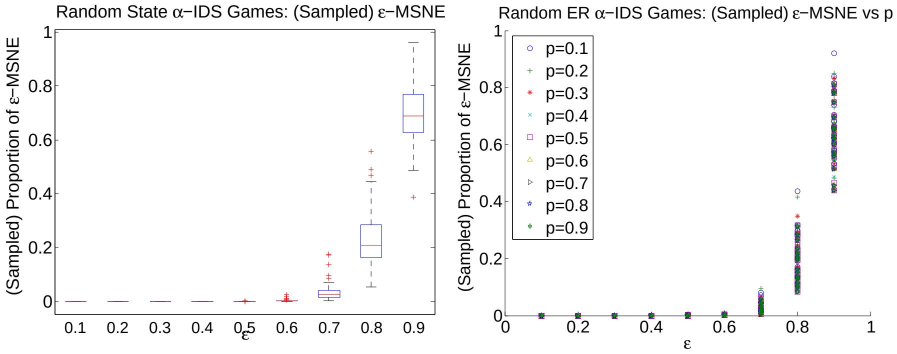


Figure 6.1: **Sampled proportional of ϵ-MSNE in random α-IDS games.** The plots show the sampled proportion of ϵ-MSNE of a fixed U.S. State topology (left) and random topology with various density (right) of random α-IDS games of 48 players. The x-axis represents the ϵ values and the y-axis represents the sampled proportion of ϵ-MSNE.

values of the parameters of α-IDS games uniformly at random between zero and one. Finally, we randomly sample 100,000 mixed-strategies and check to see how many out of the 100,000 are ϵ-MSNE for $\epsilon \in \{0.1, 0.2, ..., 0.9\}$. For the random state α-IDS games, we generate 100 of them, and for each of them we compute the sampled proportion of ϵ-MSNE for each $\epsilon \in \{0.1, 0.2, ..., 0.9\}$. Using this data, we construct the left boxplot of Figure 6.1. For this plot, we observe that as ϵ goes to zero the sampled proportion of ϵ-MSNE decreases exponentially. This suggests that the true proportion of ϵ-MSNE is very small. Similarly for the random ER α-IDS games, we consider different $p \in \{0.1, 0.2, ..., 0.9\}$, and for each fixed p, we generate 20 α-IDS games and compute the sampled proportion of ϵ-MSNE of each of them. Using this data, we construct the right boxplot of Figure 6.1. Again, we observe that as ϵ goes to zero the sampled proportion of ϵ-MSNE decreases exponentially regardless of the density and structure of the game graphs. Therefore, our empirically results would help to justify the use of our proposed method to learn the parameters of α-IDS games.

Maximizing the Number of ϵ-MSNE in the Dataset

In our approach, we subdivide the optimization by first optimizing over $\mathcal{G}$, and then optimizing over ϵ. For any ϵ, we would like to apply a simple gradient-ascent optimization technique to learn the game $\mathcal{G}$. Unfortunately, even the latter maximization is non-trivial due to the discontinuities induced by the indicator functions defining the ϵ-MSNE constraints. Our goal is then to further approximate $\widehat{\pi}^{\epsilon}(\mathcal{G})$. First, we use a simple upper bound that results from using Equation 6.3 and Equation 6.4, which correspond to satisfying the ϵ-MSNE of the games:

$$\begin{aligned}\widehat{\pi}^{\epsilon}(\mathcal{G}) &= \max_{\mathcal{G}} \frac{1}{m} \sum_{l=1}^{m} \mathbb{1}\big[x^l \in NE_{\epsilon}(\mathcal{G})\big] \\ &\leq \max_{\mathcal{G}} \frac{1}{m} \sum_{l=1}^{m} \sum_{i=1}^{n} \mathbb{1}\big[x_i^l \Delta_i^l \leq \epsilon\big] + \mathbb{1}\big[-(1-x_i^l)\Delta_i^l \leq \epsilon\big].\end{aligned}$$

Then, we approximate the indicator function in the last upper bound with another differentiable function. In the following subsection, we discuss what is perhaps the simplest approximation to the indicator function: using the logistic/sigmoid function. This is the standard approach leading to the famous BackProp algorithm used to train neural networks from data (see, e.g., the book by Haykin [1999], for more information).

Using the Logistic/Sigmoid Function

The first approximation to the upper bound above that we consider uses the sigmoid function, $s(x) \equiv \frac{1}{1+e^{-x}}$, which yields the following approximation to the last upper bound:

$$\max_{\mathcal{G}} \frac{1}{m} \sum_{l=1}^{m} \sum_{i=1}^{n} s(-x_i^l \Delta_i^l + \epsilon) + s((1-x_i^l)\Delta_i^l + \epsilon).$$

To avoid overfitting and to introduce our bias for “sparse” (graphical) game structures, we regularize the transfer parameters q_{ji}. In particular, those transfer probabilities implicitly define the structure of the α-IDS games That is, viewing α-IDS games from the perspective of a (directed, parametric) graphical game, the directed graph capturing the direct transfer risks between the players is such that each node in the graph represents a player in the game, and there is a directed edge (i.e., an arc) from node j to node i if and only if $q_{ji} > 0$. The typical regularizer used to induce sparsity in the learned structure is the L_1-regularizer, which we impose over the q_{ji}'s. We denote by $\lambda > 0$ the regularization parameter quantifying the amount of penalization for large values of the q_{ji}'s.

$$\max_{\mathcal{G}} \frac{1}{m} \sum_{l=1}^{m} \sum_{i=1}^{n} S(-x_i^l \Delta_i^l + \epsilon) + S((1 - x_i^l)\Delta_i^l + \epsilon) + \lambda \sum_{j=1}^{n} q_{ji}.$$

We "learn" λ using cross-validation. (This is the typical approach to find an "optimal" λ in ML.)

Before continuing, there is an important normalization constraint on the utility/costs functions required for the ϵ parameter for the approximation to be meaningful. In particular, recall that in order to define ϵ-MSNE, we want to ensure that the cost function of each player of α-IDS games is between zero and one for each possible mixed strategy. The following expression leads to a normalized cost function, denoted by $\widetilde{M_i}$, for player i:

$$\widetilde{M_i}(x_i, x_{-i}) \equiv \frac{M_i(x_i, x_{-i}) - \min_i}{\max_i - \min_i}$$

where $\min_i = \{C_i, p_i L_i\}$, $\max_i = \{C_i + \alpha_i r_i(0_{-i}) L_i, [p_i + (1 - p_i) r_i(0_{-i})] L_i\}$, and 0_{-i} stands for the vector that sets all the elements of x_{-i} to the value 0, so that $r_i(0_{-i}) = 1 - \prod_{j \neq i}(1 - q_{ji})$.

Notice that, if the minimum and the maximum, respectively, of the cost function of each player of α-IDS is exactly 0 and 1, respectively, then we do not have to perform any normalization when computing and evaluating ϵ-MSNE.

Unfortunately, working with the normalized costs $\widetilde{M_i}$'s is cumbersome. Instead, we keep the ϵ-MSNE constraints in terms of the original (unnormalized) cost functions M_i's and introduce additional constraints based on the expressions for $\min_i$ and $\max_i$ given above directly into the optimization problem. Using primal-dual optimization, in which we denote by the corresponding dual-variables/Lagrange-multipliers β_i and γ_i for each additional cost-function normalization for each player i, we obtain the following minimax program:

$$\begin{aligned}&\min_{\delta,\beta}\max_{\mathcal{G}}\frac{1}{m}\sum_{l=1}^{m}\sum_{i=1}^{n}S(-x_i^l\Delta_i^l+\epsilon)+S((1-x_i^l)\Delta_i^l+\epsilon)+\lambda\sum_{j=1}^{n}q_{ji}\\&-\delta_i(C_i-1)(p_iL_i-1)\\&-\gamma_i(2-(C_i+\alpha_i r_i(0_{-i})L_i))(2-([p_i+(1-p_i)r_i(0_{-i})]L_i))\ ,\end{aligned}\tag{6.5}$$

where $\beta = (\beta_1, ..., \beta_n)$ and $\gamma = (\gamma_1, .., \gamma_n)$ We intentionally enforce that $\min_i = \{C_i, p_i L_i\} = 1$ and $\max_i = \{C_i + \alpha_i r_i(0_{-i})L_i, [p_i + (1-p_i)r_i(0_{-i})]L_i\} = 2$ to avoid computational issues. As long as the difference of the $\min_i$ and $\max_i$ is close to 1, then we can easily see that the ϵ-MSNE definition will be well-defined. We want to solve the above program subject to the respective constraints on each of the variables. As stated previously, we follow the traditional approach of using gradient-ascent/descent optimization as a heuristic to update, and eventually learn, the parameters.

Denote by $(q', \mathcal{G}', \epsilon')$ the tuple we learn using the approach we propose above. In this paper we assume that $\mathcal{NE}^{\epsilon'}(\mathcal{G}')$ is measurable, so that the learned generative model is well-defined.

6.4 Preliminary Experiment

As mentioned in the introduction, we will use the publicly-available CDC state-level vaccination-rate data to learn an α-IDS game. In particular, we will be using the 2009-2010 US states H1N1 vaccination percentages and their standard deviations as shown below for a few US states.

Figure 6.2 shows the choropleth map of the percentage (darker colors mean higher percentages) of sampled population in each state of the U.S.A continental (i.e., excluding Alaska and Hawaii) that reported taking the H1N1 vaccine in the year-period of 2009-2010. The vaccination percentages range from 17.5% to 46.8% with MS and RI with the lowest and highest percentages, respectively. Table 6.3 shows the vaccination percentages and the standard deviations for some of the states in the US that are used to generate Figure 6.2.

Viewing each state as a player in the network, we interpret the vaccination percentages as mixed-strategies and generate m samples i.i.d according to an n-variate product-of-normal distribution, where $n = 48$ in our case, with the joint mean and standard deviations given by each state's

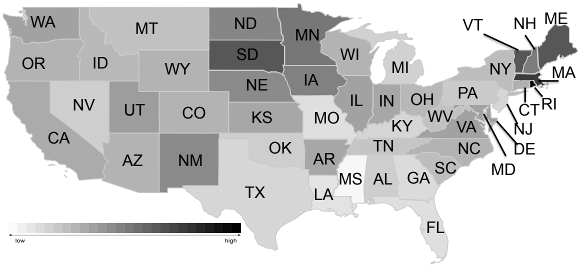


Figure 6.2: **Choropleth map of vaccination percentage.** This is the choropleth map of percentage of sample population who reportedly took the H1N1 vaccine in the US continent during (May) 2009 - 2010. Darker regions correspond to higher vaccination percentage.

	% H1N1	% Standard
State	Vaccinated	Deviation
MS	17.5%	1.5%
LA	21.1%	1.7%
OK	24.0%	1.6%
AR	29.3%	4.3%
NM	33.2%	2.6%
RI	46.8%	2.0%
TX	22.7%	1.5%

Figure 6.3: **State vaccination percentage** The columns of the table corresponds to some of the states, their vaccination percentages, and their standard deviations, respectively from left to right, as used to construct Figure 1.

reported vaccination-rate and standard deviation in the CDC data, where $m \in \{500, 1000, 1500, 2000, 2500, 5000\}$. The second and third column in Table 6.3 provide examples for the corresponding mean and standard deviation of some states, where the information in each row corresponds to that of the respective individual example state listed in the first column of the table. It is important to note that we do not have publicly available information about any (higher-level) correlations among states in the CDC data. Hence, each one of the m samples is a joint mixed-strategy of dimension $n = 48$, and each component in the joint-mixed-strategy sample is drawn independently according to the mean and variance of the respective state in the continental US, as reported in the CDC data.

We impose an *a priori* bias for learning where only neighboring states may transfer the virus. Therefore, we are restricting ourself to learning a geospatially-informed continental-US-state-level α-IDS game. To actually learn the values of the parameters of an α-IDS game, we take partial derivatives of Equation 6.5 with respect to the parameters C_i, L_i, α_i, p_i, and $(q_{ji})_{j \in \mathrm{Pa}(i)}$ for each player i and use the standard gradient-ascent optimization technique. The process terminates when the cost functions are normalized (i.e., $\min_i = \{C_i, p_i L_i\} = 1$ and $\max_i = \{C_i + \alpha_i r_i(0_{-i}) L_i, [p_i + (1 - p_i) r_i(0_{-i})] L_i\} = 2$ for every i) and after exceeding some threshold on the maximum number of iterations.

Moreover, we experiment with different regularization parameter values of β, δ, λ, and ϵ and with various sample sizes.

In what follows, we present our learned α-IDS game with $\beta = \delta = -2$, $\lambda = 1$, $\epsilon = 0.35$, and $n = 1500$ which we found through empirical observations and cross-validation achieved the best log-likelihood for the CDC H1N1 vaccination dataset.

6.4.1 Parameters of the Players of Learned α-IDS game

We begin by discussing the values of the parameters we learned for the players in the learned α-IDS game from the CDC vaccination dataset.

Players' Characteristics

The first thing to note is each player's type. Recall that there are two types of players in an α-IDS game, whose characterization of best-response behavior is to exhibit either strategic complementarily (SC) or strategic substitutability

(SS). If the player is SC, then the player will play the action vaccinate if there are “enough” of his/her neighbors play the action vaccinate. On the other hand, if the player is SS, then the player will play the action vaccinate if not “enough” of his/her neighbors play the action vaccinate. Said differently, if a reasonable amount of the player’s neighbor vaccinate, then the SS player will not.

Indeed, in the vaccination setting, intuition suggest that one would expect *all* players to be SS; there is no reason for his/her to vaccinate if enough neighboring players around the player are protected from the virus. *In fact, all of the players we learned are SS.* Figure 6.4 shows exactly this. Recall that in the α-IDS game, to determine whether a player is SC or SS, we only need to compare the player’s α and $1 - p$ values. If $\alpha > 1 - p$, then the player’s type is SC. If $\alpha < 1 - p$, the the player’s type is SS. In Figure 6.4, we plot the α and $1 - p$ values of each player where the α values are on the x-axis and the $1 - p$ values are on the y-axis. The line denotes the values at which $\alpha = 1 - p$. Notice that the line corresponding to $\alpha = 1 - p$ is not shown as a straight diagonal because we draw the x- and y-axis to different scales. We zoom in intentionally to the portion of the plot that contains the data.

As observed from the figure, all of the points are above the line. This indicates that all of the players are the type of SS (with $1 - p_i > \alpha_i$ for all player i).

It is important to note that there is no reason, a priori, as to why our learning formulation and algorithms/heuristics would yield models in which all players turned out to be SS. Although, a posteriori, that is the most natural observation/result, consistent with our general intuition/expectation about vaccination scenarios. Thus, the results presented in the plot provide some partial, anecdotal, and favorable evidence that the game we learned is not arbitrary.

Best-response Correspondences of the Players

Recall that to determine the best-response of a player, we look at his/her’s best-response correspondence. In particular, the best-response correspon-

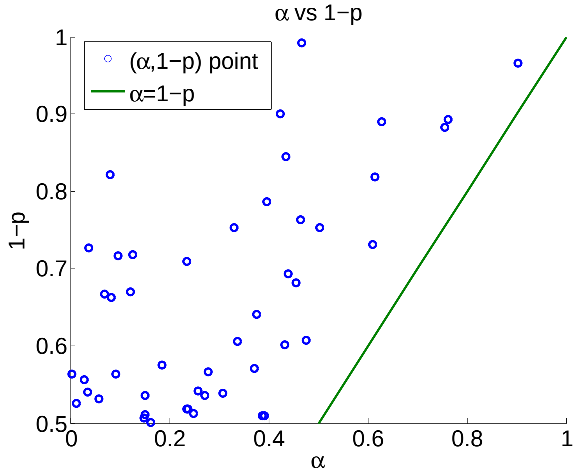


Figure 6.4: **Players' Types.** The x-axis denotes the α values of the players, the y-axis denotes the $1-p$ values of they players, and the line is the equation $\alpha = 1 - p$. The plot is scaled to capture the α and $1 - p$ values. The plot illustrates that our learning formulation produces values of the parameters that are consistent with vaccination scenarios.

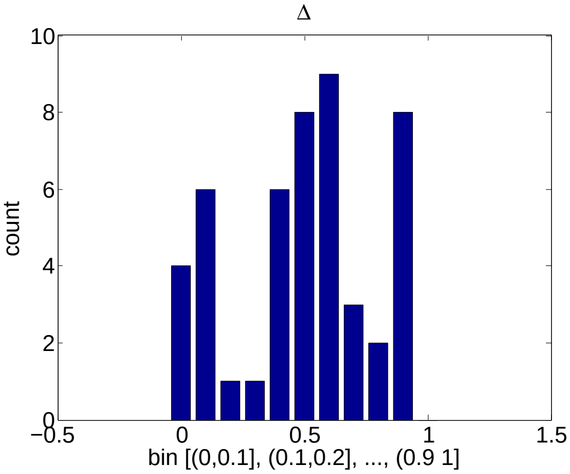


Figure 6.5: **Histograms of the Δ values.** x-axis is discretized into the range of $(0, 0.1], (0.1, 0.2], ..., (0.9, 1]$ and y-axis is the count of Δ values in the ranges.

dence of a SS player i is

$$\mathcal{BR}_i^{ss}(a_{\mathrm{Pa}(i)}) \equiv \begin{cases} \{0\}, & \Delta_i^{ss} < s_i(a_{\mathrm{Pa}(i)}), \\ \{1\}, & \Delta_i^{ss} > s_i(a_{\mathrm{Pa}(i)}), \\ \{0,1\}, & \Delta_i^{ss} = s_i(a_{\mathrm{Pa}(i)}) \ , \end{cases}$$

where $\Delta_i^{ss} = 1 - \frac{\frac{C_i}{Li} - p_i}{1 - pi - \alpha_i}$. In order for player i to have a non-trivial response, the value of Δ_i^{ss} has to be between zero and one. Indeed, in our learned IDS games, the Δ_i^{ss} for all players i is between zero and one. Figure 6.5 shows a histogram of the Δ_i values of each player i. The values fall roughly between the range of $(0.010, 0.999)$.

Transfer Risks of the Players

Recall that the transfer risks of a player are the $(q_{ji})_{j \in \mathrm{Pa}(i)}$ where q_{ji} is the probability that a virus will transfer from j to i. Of course, our learned

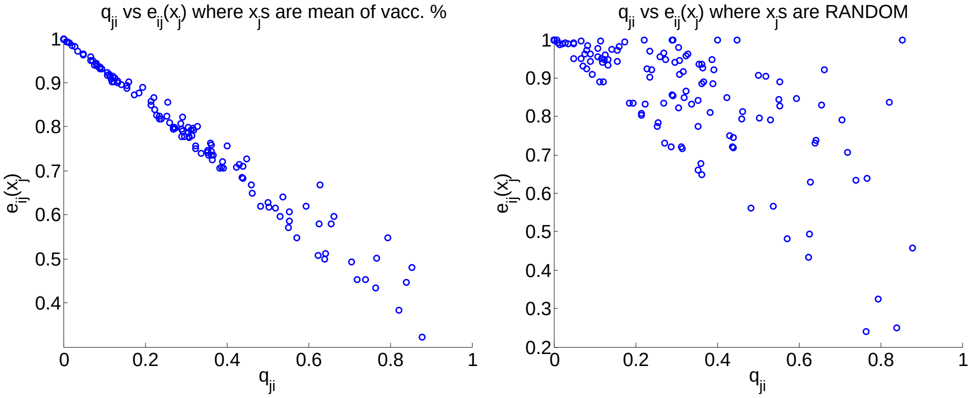


Figure 6.6: **Values of the Safety Functions.** The safety functions are evaluated using the mean of vaccination (Left) and using a random mixed-strategy (Right). The x-axis represents the transfer risks and the y-axis represents the values of the safety functions.

transfer risks depend on the mixed-strategies of the players that we use to learn the values. To show that our learned transfer risks is consistent with the training examples, we compute the safety values of each player from his neighbors using the vaccination-rate data (the mean rate we used to generate the examples). More specifically, we compute $e_{ji} = x_j + (1 - x_j)(1 - q_{ji})$ for each i and $j \in \text{Pa}(i)$. We also compare the values of the e_{ji} to those of values using some random mixed-strategies. The results are shown in Figure 6.6.

In Figure 6.6, we plot the q_{ji} and its corresponding e_{ji} values given the mean vaccination-rate (left) and a random mixed-strategy (right). The left plot shows an obvious regularity not observed on the right plot. This suggests that the transfer risks that we learned are not random and correlated to the training examples. Hence, the results presented in Figure 6.6 provide another piece of evidence suggesting that our learned models are not arbitrary, and that, on the contrary, they seem consistent with our general intuition regarding real-world vaccination settings

Structure of the Graph

The magnitude of the learned transfer-risk parameters determine the structure of the underlying game-graph induced from the data. Figure 6.7 shows the learned game-graph in full (Top) and a zoomed in portion of the New England area (Bottom). The nodes are the 48 US states excluding AK and HI. The directed edges denote the transfer risks from a state to another state. For instance, in the zoom-in version (Bottom), the probability that NY can transfer the virus to MA is 0.142.

Equilibrium Behavior of the Players

Our main interest for learning games is the ability they provide to potentially interpret what would happen at an MSNE, even when the given data may not consists of all examples being exact MSNE, or may be noisy. Said differently, the mixed-strategies of the state agents in our data may not correspond to the "optimal" equilibrium strategies, by which we mean exact MSNE strategies.

In short, we want to infer and study the behavior of the players (i.e., US states), at an exact or approximate MSNE of the learned game model, from noisy data, in which not all examples may belong to the set of ϵ-MSNE of some fixed but unknown game. Thus, given the learned games, we can run a version of some learning-heuristics/regret-minimization [Fudenberg and Levine, 1998], in which we use the mean vaccination rates as the initial mixed-strategy profile to compute ϵ-MSNE in these games.

Figure 6.8 shows the ϵ-MSNE we obtain after the best-response-gradient dynamics converges, whenever it converges for $\epsilon \in \{0.35, 0.15, 0.05, 0\}$. It turns out that the mean vaccination-rates given in the CDC data is an 0.35-MSNE of the learned game. Note that this observation is non-trivial because there is no technical *a priori* reason to expect such a result: there is nothing in our learning algorithm that enforces any such condition, and the data might have as well led our learning algorithms to yield games for which such mean vaccination-rates might not have been an 0.35-MSNE of the learned game. Moreover, we are able to find an exact MSNE which is also a PSNE after trying many initial mixed-strategies that are drawn uniformly at random for the learning heuristic. *A posteriori*, it is somewhat reassuring to find "free-riders" at MSNE of the learned games, which is again consistent with the expectations of the behavior of players in vaccination-type settings. For instance, according to our learned model, at an equilibrium, NH plays the

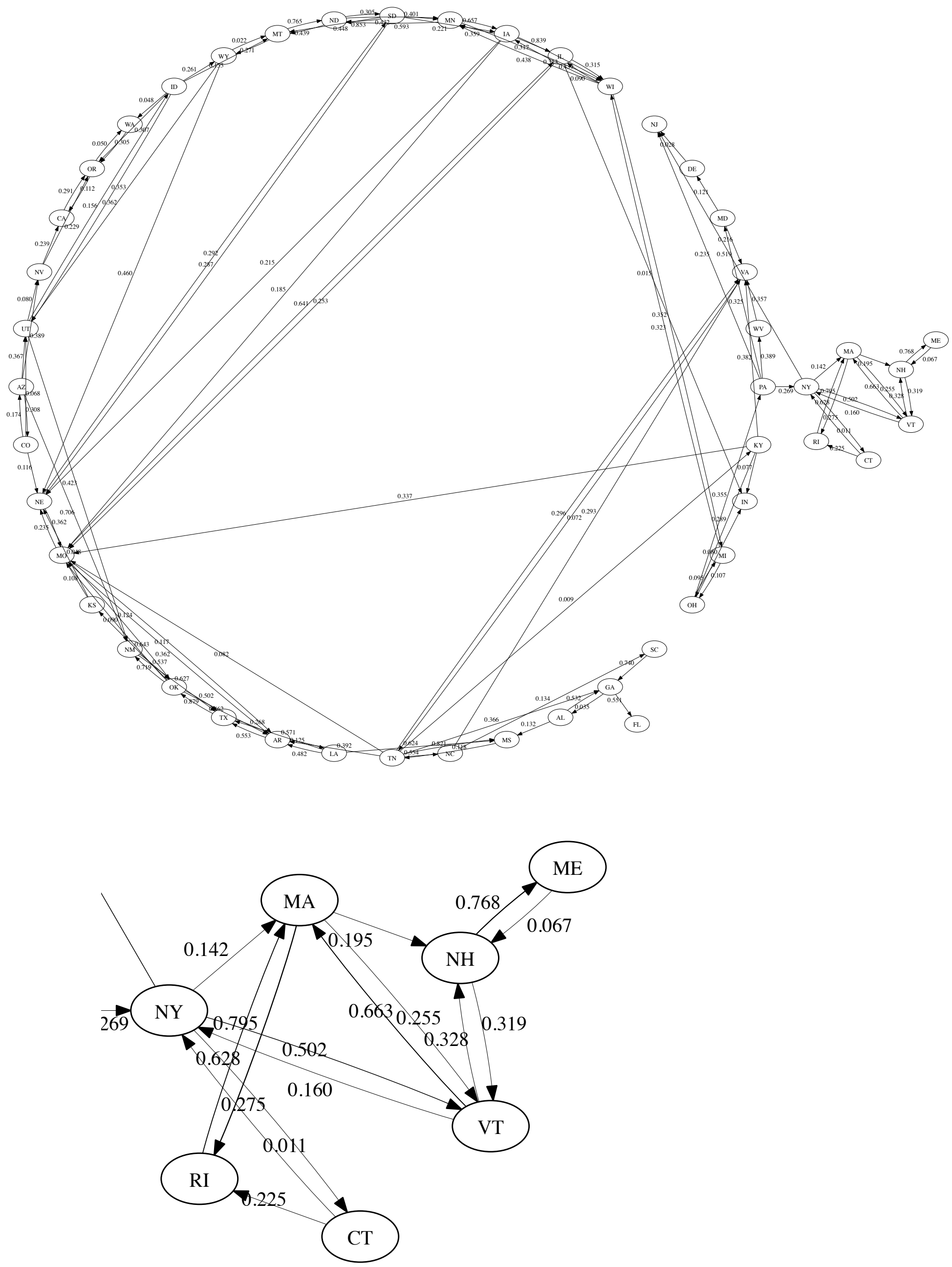
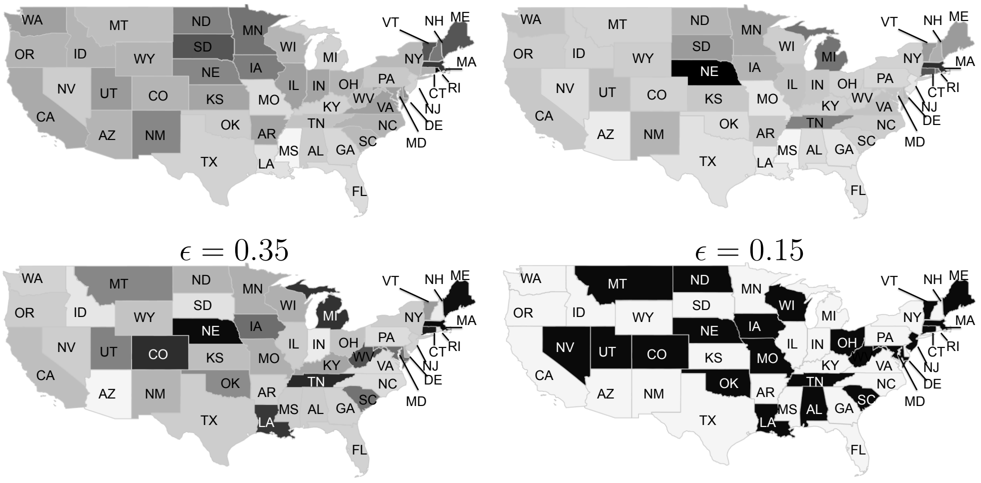


Figure 6.7: **Learned game-graph.** The nodes are the 48 US states (excluding AK and HI) and the (directed) edges denote the transfer risks from the source to destination. Top = Full Graph, Bottom = Zoom-in portion of the New England Area.



$\epsilon = 0.35$ $\epsilon = 0.15$

$\epsilon = 0.05$ exact MSNE

Figure 6.8: **Equilibrium of the Learned α-IDS game.** The ϵ-MSNE to which best-response-gradient dynamics consistently converged for $\epsilon \in \{0.35, 0.15, 0.05, 0\}$. Darker regions correspond to higher probability of vaccination (i.e., vaccination rates), for the respective ϵ-MSNE

action of not vaccinate while all of its neighbors vaccinate; we can see a similar situation for KS.

6.5 Conclusion

In this chapter, we propose and discuss a new learning problem to learn parameter values for game-models to capture and compactly represent approximate MSNE from mixed-strategy-based data. In particular, we first propose a specific, simple generative model of mixed-strategy-based data, which should serve as a starting point for future, potentially more sophisticated models for behavioral data of such strategic nature, and how it might have been collected or generated.

Given the generative model, we then propose a specific way to learn game-model parameters with the objective of capturing and compactly representing (approximate) MSNE embedded within mixed-strategy-based behavioral datasets. As a particular instance of our learning framework, we propose a specific way to learn α-IDS games.

To illustrate the effectiveness of our proposed framework and methodology, we present the results of preliminary experiments to learn and study models inferred from real-world, publicly-available data, the CDC dataset on vaccination-rates for the continental US. Notwithstanding the preliminary nature of our experimental work, our experimental results show that the learned parameters are consistent with our intuitive understanding of both the local and global system behavior one would expect from data collected in vaccination settings.

Of course, while our preliminary results are promising, we still need a more thorough experimental evaluation for proper validation of the overall effectiveness of our proposed machine-learning framework and methodology, including our biases for model selection and our learning algorithms/heuristics.

Chapter 7

Conclusion

My doctoral thesis consists of the following components: (1) designing increasingly more realistic variants of defense games; (2) studying computational questions in defense games such as equilibria computation and computational implications of equilibria characterizations, (3) designing efficient algorithms and effective heuristics for defense problems; and (4) designing and applying new machine learning techniques to estimate game model parameters from behavioral data.

In particular, we first introduce α-IDS games to study the settings (i.e., airline security, fire protection, and vaccination) where each individual's investment can partially protect transfer risks from others. We study the computational complexity of computing NE in various classes of α-IDS games and show that computing a PSNE in general α-IDS games is NP-complete. For some instances of the games, we introduce efficient algorithms to compute all NE for that instances. We then perform experiments to show the behavior of the players in the games at ϵ-MSNE.

Next, we build from the α-IDS games and introduce IDD games that model the present of an attacker who deliberately wants to cause harm to the system. We focus on the case where there is only one attack. We show that there is no PSNE in any IDD games and there is an efficient algorithm to compute all NE in a class of IDD games. Moreover, we investigate the question of computing ϵ-MSNE in IDD games and show that there is an FPTAS to compute an ϵ-MSNE in directed tree structures. We perform a series of experiments to show the behavior of the attacker and the players/sites/defenders in the system at ϵ-MSNE.

Finally, we study the question of learning the parameters of the games using mixed-strategies. We provide a simple and general machine learning framework to learn the parameters of any games given mixed-strategies. As an application, we apply the framework and use some machine learning techniques to learn the parameters of α-IDS games given the CDC vaccination data. Our experimental results show that the learned parameters are consistent with our intuitive understanding of both the local and global system behavior one would expect from data collected in vaccination settings.

Of course, this is just the beginning and there are quite a few open problems such as the complexity of computing PSNE and ϵ-MSNE in some classes of α-IDS games and the complexity of computing ϵ-MSNE in general IDD games. There are also some open questions in regard to modeling and studying IDD games in the settings where there are multiple attackers and the attacker(s) has (have) more than one attack. There are also some open problems in regard to the work of learning the parameters of games from mixed-strategies. Even through the preliminary results are promising, we still need a more thorough experimental evaluation for proper validation of the overall effectiveness of our proposed machine-learning framework and methodology, including our biases for model selection and our learning algorithms/heuristics. We refer the readers to the respective conclusion section of the chapters for more detail in regard to the open problems.

Bibliography

S. Agiwal and H. Mohtadi. Risk Mitigating Strategies in the Food Supply Chain. *American Agricultural Economics Assocation (Annual Meeting)*, July 2008.

R. E. Bellman. *Dynamic Programming.* Dover Publications, Incorporated, 2003.

V. M. Bier and M. N. Azaiez, editors. *Game Theoretic Risk Analysis of Security Threats.* Springer, 2009.

E. Cárceles-Poveda and Y. Tauman. Strategic Aspects of Terrorism. *Games and Economic Behavior*, 2011. Forthcoming.

X. Chen and X. Deng. Settling the Complexity of Two-Player Nash Equilibrium. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, 2006.

X. Chen, X. Deng, and S. Teng. Settling the Complexity of Computing Two-player Nash Equilibria. *J. ACM*, 56(3):14:1–14:57, May 2009.

V. Conitzer and T. Sandholm. New Complexity Results about Nash Equilibria. *Games and Economic Behavior*, 63(2):621 – 641, 2008.

C. Daskalakis and C. H. Papadimitriou. Three-Player Games Are Hard. Technical Report 139, Electronic Colloquium on Computational Complexity (ECCC), 2005.

C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. The Complexity of Computing a Nash Equilibrium. *Commun. ACM*, 52(2):89–97, 2009.

R. Dechter. *Constraint Processing.* Morgan Kaufmann Publishers Inc., 2003.

Q. Duong, Y. Vorobeychik, S. Singh, and M. P. Wellman. Learning Graphical Game Models. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI'09, pages 116–121, San Francisco, CA, USA, 2009.

Q. Duong, M. P. Wellman, S. Singh, and M. Kearns. Learning and Predicting Dynamic Networked Behavior with Graphical Multiagent Models. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '12, pages 441–448, 2012.

E. Elkind, L. A. Goldberg, and P. W. Goldberg. Nash Equilibria in Graphical Games on Trees Revisited. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, EC '06, pages 100–109, 2006.

P. Erdös and A. Rényi. On random graphs, I. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

S. Ficici, D. Parkes, and A. Pfeffer. Learning and Solving Many-Player Games through a Cluster-Based Representation. In *Proceedings of the Twenty-Fourth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-08)*, pages 188–195, 2008.

D. Fudenberg and D. K. Levine. *The Theory of Learning in Games*, volume 1 of *MIT Press Books*. MIT Press, June 1998.

D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, Cambridge, MA, 1991.

X. A. Gao and A. Pfeffer. Learning Game Representations from Data Using Rationality Constraints. In *Proceedings of the 26th Conference on on Uncertainty in Artificial Intelligence (UAI'10)*, 2010.

M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12): 7821–7826, 2002.

K. Gkonis and H. Psaraftis. Container transportation as an interdependent security problem. *Journal of Transportation Security*, 3:197–211, 2010.

G. Gottlob, G. Greco, and F. Scarcello. Pure Nash Equilibria: Hard and Easy Games. *J. Artif. Int. Res.*, 24(1):357–406, 2005.

S. Haykin. *Neural Networks: A Comprehensive Foundation.* Prentice Hall, 1999.

G. Heal and H. Kunreuther. You Only Die Once: Managing Discrete Interdependent Risks. Working Paper 9885, National Bureau of Economic Research, August 2003.

G. Heal and H. Kunreuther. Interdependent Security: A General Model. Working Paper 10706, National Bureau of Economic Research, August 2004.

G. Heal and H. Kunreuther. IDS Models of Airline Security. *Journal of Conflict Resolution*, 49(2):201–217, April 2005a.

G. Heal and H. Kunreuther. The Vaccination Game. Working paper, Wharton Risk Management and Decision Processes Center, January 2005b.

G. Heal and H. Kunreuther. Modeling Interdependent Risks. *Risk Analysis*, 27:621–634, July 2007.

J. Honorio and L. E. Ortiz. Learning the Structure and Parameters of Large-Population Graphical Games from Behavioral Data. *Journal of Machine Learning Research.*, (In Press), 2014.

M. T. Irfan and L. E. Ortiz. On Influence, Stable Behavior, and the Most Influential Individuals in Networks: A Game-Theoretic Approach. *Artificial Intelligence*, 215:79–119, 2014.

M. Jain, D. Korzhyk, O. Vaněk, V. Conitzer, M. Pěchouček, and M. Tambe. A Double Oracle Algorithm for Zero-sum Security Games on Graphs. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '11, pages 327–334, 2011.

M. Kearns. Economics, Computer Science, and Policy. *Issues in Science and Technology*, Winter 2005.

M. Kearns. Graphical games. In Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay V. Vaziran, editors, *Algorithmic Game Theory*, chapter 7, pages 159–180. Cambridge University Press, 2007.

M. Kearns and L. E. Ortiz. Algorithms for Interdependent Security Games. In *Advances in Neural Information Processing Systems*, NIPS '04, pages 561–568, 2004.

M. Kearns and J. Wortman. Learning from Collective Behavior. In *In Proceedings of the 21st Annual Conference on Learning Theory*, COLT '04, pages 99–110, 2008.

M. Kearns, M. Littman, and S. Singh. Graphical Models for Game Theory. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, UAI '01, pages 253–260, 2001.

C. Kiekintveld, M. Jain, J. Tsai, J. Pita, F. Ordóñez, and M. Tambe. Computing Optimal Randomized Resource Allocations for Massive Security Games. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '09, pages 689–696, 2009.

Jon Kleinberg. Cascading Behavior in Networks: Algorithmic and Economic Issues. In Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay V. Vazirani, editors, *Algorithmic Game Theory*, chapter 24, pages 613–632. Cambridge University Press, 2007.

B. Klimt and Y. Yang. Introducing the Enron Corpus. In *CEAS*, 2004.

D. E. Knuth. *The Stanford GraphBase: A Platform for Combinatorial Computing*. ACM, 1993.

D. Korzhyk, V. Conitzer, and R. Parr. Complexity of Computing Optimal Stackelberg Strategies in Security Resource Allocation Games. In *Proceedings of the National Conference on Artificial Intelligence*, AAAI '10, pages 805–810, 2010.

D. Korzhyk, V. Conitzer, and R. Parr. Security Games with Multiple Attacker Resources. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume One*, IJCAI'11, pages 273–279, 2011a.

D. Korzhyk, V. Conitzer, and R. Parr. Solving Stackelberg Games with Uncertain Observability. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 3*, AAMAS '11, pages 1013–1020, 2011b.

H. Kunreuther and G. Heal. Interdependent Security. *Journal of Risk and Uncertainty*, 26(2-3):231–249, March 2003.

A. Laszka, M. Felegyhazi, and L. Buttyan. A Survey of Interdependent Information Security Games. *ACM Comput. Surv.*, 47(2):23:1–23:38, August 2014.

J. Leskovec, D. Huttenlocher, and J. Kleinberg. Signed Networks in Social Media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1361–1370, 2010.

P. Liu. Incentive-Based Modeling and Inference of Attacker Intent, Objectives, and Strategies. In *Proc. of the 10th ACM Computer and Communications Security Conference (CCS'03)*, pages 179–189, 2003.

K. W. Lye and J. Wing. Game Strategies in Network Security. In *Proceedings of the Workshop on Foundations of Computer Security*, pages 1–2, 2002.

J. Nash. Equilibrium Points in n-Person Games. *Proceedings of the National Academy of Sciences of the United States of America*, 35(1):48–49, Jan. 1950.

J. Nash. Non-Cooperative Games. *Annals of Mathematics*, 54:286–295, September 1951.

N. Nisan, T. Roughgarden, É. Tardos, and V. V. Vazirani, editors. *Algorithmic Game Theory.* Cambridge University Press, 2007.

A. O'Connor and E. Schmitt. Terror Attempt Seen as Man Tries to Ignite Device on Jet. The New York Times, 25 December 2009. Cited 31 August 2010. Avaliable at `http://www.nytimes.com/2009/12/26/us/26plane.html`.

L. E. Ortiz. On Sparse Discretization for Graphical Games. Technical report, arXiv:1411.3320 [cs.AI], 2014.

T. E. S. Raghaven, T. S. Ferguson, T. Parthasarathy, and O.J. Vrieze. *Stochastic Games And Related Topics: In Honor of Professor L. S. Shapley.* Theory and Decision Library C. Springer Netherlands, 1990.

S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya, and Q. Wu. A Survey of Game Theory as Applied to Network Security. In *Proceedings of the 2010 43rd Hawaii International Conference on System Sciences*, HICSS '10, pages 1–10, 2010.

Y. Shavitt and E. Shir. DIMES: Let the Internet Measure Itself. *ACM SIGCOMM Computer Communication Review*, 35(5):71–74, October 2005.

Y. Shoham and Kevin Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations.* Cambridge University Press, Cambridge, UK, 2009.

S. Singh, M. Kearns, and Y. Mansour. Nash Convergence of Gradient Dynamics in General-Sum Games. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, UAI '00, pages 541–548, 2000.

D. M. Topkis. Equilibrium Points in Nonzero-Sum n-Person Submodular Games. *SIAM Journal on Control and Optimization*, 17(6):773–787, 1979.

V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag New York, Inc., 2001.

X. Vives. Nash equilibrium with strategic complementarities. *Journal of Mathematical Economics*, 19(3):305–321, 1990.

J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior.* Princeton University Press, 1944. Second Edition, 1947.

Y. Vorobeychik, M. P. Wellman, and S. Singh. Learning Payoff Functions in Infinite Games. *Mach. Learn.*, 67(1-2):145–168, May 2007.

D. J. Watts and S. H. Strogatz. Collective Dynamics of Small-World Networks. *Nature*, 393:440–442, 1998.

J. Wright and K. Leyton-Brown. Beyond Equilibrium: Predicting Human Behavior in Normal-Form Games. In *Proceedings of the National Conference on Artificial Intelligence*, AAAI '10, pages 901–907, 2010.

J. Wright and K. Leyton-Brown. Behavioral Game Theoretic Models: A Bayesian Framework for Parameter Analysis. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '12, pages 921–930, 2012.

W.W. Zachary. An Information Flow Model for Conflict and Fission in Small Groups. *Journal of Anthropological Research*, 33:452–473, 1977.

B. D. Ziebart, J. A. Bagnell, and A. K. Dey. Modeling Interaction via the Principle of Maximum Causal Entropy. In *Proceedings of the 27th International Conference on Machine Learning*, ICML '10, pages 1255–1262, 2010.