Empirical Game-Theoretic Methods for Strategy Design and Analysis in Complex Games

by

Christopher D. Kiekintveld

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in The University of Michigan
2008

Doctoral Committee:
Professor Michael P. Wellman, Chair
Professor Yan Chen
Professor Jeffrey MacKie-Mason
Associate Professor Satinder Singh Baveja

Table of Contents

List of Tables iv

List of Figures vi

Abstract viii

Chapter 1 Introduction to Games and Empirical Game Modeling 1
- 1.1 Game Definitions and Terminology 4
  - 1.1.1 Normal-Form Games 4
  - 1.1.2 Bayesian Games 5
- 1.2 Solution Concepts for Games 6
  - 1.2.1 Dominance 7
  - 1.2.2 Nash Equilibrium 7
- 1.3 Empirical Models of Games 10
- 1.4 Thesis Overview 14

Chapter 2 Empirical Game-Theoretic Analysis Applied to the TAC Supply-Chain Game 16
- 2.1 The TAC Supply Chain Management Game 17
  - 2.1.1 Game Description 17
  - 2.1.2 Strategic Interactions in TAC SCM 19
  - 2.1.3 Challenges for Strategic Analysis 20
- 2.2 Empirical Game Modeling for TAC SCM 21
  - 2.2.1 Reducing the Game 23
  - 2.2.2 Estimating the Payoff Matrix 24
- 2.3 Game-Theoretic Analysis of TAC SCM 25
  - 2.3.1 Analysis of SCM 2006 Agents 26
  - 2.3.2 Analysis of Combined 2005 and 2006 Agents 30
  - 2.3.3 Discussion 33
- 2.4 Related Work 34

Chapter 3 Strategy Selection for Empirical Games: The Meta-Game Model . . 37
- 3.1 The Meta-Game Model 39
  - 3.1.1 Formal Specification 39

3.1.2 Discussion . . . 43
3.1.3 Meta-Games with Directed Observations . . . 45
3.2 Empirical Evaluation of Meta-Strategies . . . 46
3.2.1 Challenges for Finding Exact Solutions to Meta-Games . . . 48
3.2.2 Experimental Framework . . . 49
3.3 Naïve Equilibrium Analysis . . . 51
3.4 Related Work . . . 54

Chapter 4 Selecting Strategies with Noisy Game Observations . . . 57
4.1 Candidate Meta-Strategies . . . 58
4.1.1 Pure-Strategy Approximate Nash Equilibrium . . . 60
4.1.2 Mixed-Strategy Approximate Nash Equilibrium . . . 62
4.1.3 Logit Equilibrium . . . 63
4.2 Game Classes and Observation Models . . . 65
4.3 Noisy Observation and Distributional Predictions . . . 67
4.4 Benchmarking the Meta-Strategies . . . 70
4.5 Discussion . . . 80
4.6 Related Work . . . 81

Chapter 5 Exploration Policies for Large Games . . . 84
5.1 Chaturanga . . . 85
5.1.1 A Parameterized Strategy Space for Chaturanga . . . 86
5.1.2 Strategic Independence in Chaturanga . . . 89
5.1.3 Experiments . . . 94
5.2 Candidate Exploration Policies . . . 99
5.2.1 Random Selection . . . 99
5.2.2 Tabu Best-Response Dynamics . . . 100
5.2.3 Minimum-Regret-First Search . . . 102
5.2.4 Subgame Best-Response Dynamics . . . 103
5.3 Game Classes . . . 106
5.4 Identifying Equilibrium . . . 109
5.5 Selecting Strategies . . . 114
5.6 Discussion . . . 116
5.7 Related Work . . . 120

Chapter 6 Conclusion . . . 121
6.1 Summary of Contributions . . . 122
6.2 Future Work . . . 124

Bibliography . . . 127

List of Tables

Table

1.1 An example of a matrix representation of a normal-form game. This game is a Prisoner's Dilemma, one of the most famous examples in game theory. . . 5

1.2 Example game 2, in normal form. 7

2.1 TAC SCM-06 finalists, with average scores ($M) from seeding through final rounds (semi-final and quarter-final groups in brackets). 27

2.2 Deviation statistics for the game containing SCM 2006 tournament agents. . 28

2.3 Deviation benefit comparison for top 2006 agents in the context of a symmetric mixed Nash equilibrium of top 2005 agents, using both 2005 and 2006 server rules. 31

2.4 Ranking of eleven TAC SCM agents based on deviations from an equilibrium context, coupled with tournament results (in $M). 32

3.1 An example game in the normal form, where M represents a large number. . 44

3.2 Meta-game for NPSNE and BRU with noise standard deviation of 0.05. Only the distinct payoffs are shown in the matrix. 53

4.1 Summary of meta-strategy families and their associated free parameters. . . 67

4.2 Meta-strategy parameter settings used in the experimental setup. 69

5.1 Results of regressions for a player's score and the attack angle weights placed on the player. 95

5.2 Pure-strategy profiles with minimum regret in the attack angle strategy space. 95

5.3 Statistics summarizing the data for the attack angle strategy space. Score is the average score over all profiles. Dev Benefit is the average benefit for deviating from the strategy to any other strategy. % Positive Dev is the percentage of deviations from this strategy that result in a net gain. The top X% columns show how often (as a percentage) each strategy appears in the given fraction of the pure-strategy profiles with minimum regret. 96

5.4 Strategies in the evaluation/search strategy space. A search ply represents a single move by a single player. 97

5.5 Pure-strategy profiles with minimum regret in the evaluation/search strategy space 97

5.6 Statistics summarizing the data for the evaluation/search strategy space. Score is the average score over all profiles. Dev Benefit is the average benefit for deviating from the strategy to any other strategy. % Positive Dev is the percentage of deviations from this strategy that result in a net gain. The top X% columns show how often (as a percentage) each strategy appears in the given fraction of the pure-strategy profiles with minimum regret. . . . 98

5.7 **TBRD medium games.** The average number of profiles evaluated by variations of TBRD before a Nash equilibrium is confirmed. The three parameters specify the aspiration level, minimum deviations tested, and evaluation density, respectively. The first line listed corresponds to the base TBRD algorithm. . . . 110

5.8 **TBRD large games.** The average number of profiles evaluated by variations of TBRD before a Nash equilibrium is confirmed. The three parameters specify the aspiration level, minimum deviations tested, and evaluation density, respectively. The first line listed corresponds to the base TBRD algorithm. . . . 111

5.9 **MRFS medium games.** The average number of profiles evaluated by variations of MRFS before a Nash equilibrium is confirmed. The parameter setting specifies the evaluation density for the average payoff heuristic. The first setting is the baseline random ordering. . . . 111

5.10 **MRFS large games.** The average number of profiles evaluated by variations of MRFS before a Nash equilibrium is confirmed. The parameter setting specifies the evaluation density for the average payoff heuristic; the first setting is the baseline random ordering. . . . 112

5.11 **SBRD medium games.** The average number of profiles evaluated by variations of SBRD before a Nash equilibrium is confirmed. The first parameter is the initial size of the subgame, and the second is the number iterations before the size is incremented. The first line listed corresponds to the base SBRD algorithm. . . . 112

5.12 **SBRD large games.** The average number of profiles evaluated by variations of SBRD before a Nash equilibrium is confirmed. The first parameter is the initial size of the subgame, and the second is the number iterations before the size is incremented. The first line listed corresponds to the base SBRD algorithm. . . . 112

5.13 **All exploration policies, medium games.** The average number of profiles evaluated to confirm a Nash equilibrium for each of the algorithms, taking the minimum over all parameter settings for each class of games. . . . 113

5.14 **All exploration policies, large games.** The average number of profiles evaluated to confirm a Nash equilibrium for each of the algorithms, taking the minimum over all parameter settings for each class of games. . . . 113

List of Figures

Figure

1.1 Steps in generating and analyzing an empirical game model. 12
2.1 Diagram overviewing the configuration of the TAC SCM game. 18
2.2 Deviation and profile stability analysis for the 2006 tournament agents. . . . 28
3.1 A depiction of a meta-game showing selection of the base game, generation of empirical games, and strategy selections based on the empirical games. Here, player 1 chooses strategy B and receives payoff 3, while player 2 chooses strategy D and receives payoff 9. 40
3.2 Expected payoffs for NPSNE and BRU with varying noise levels, for games with uniform random payoffs and a Gaussian observation model. 54
4.1 How changes in the parameter setting affect the LES predicted distribution over profiles. As the parameter λ is decreased, the predicted outcome distributes weight more evenly across the pure-strategy profiles. Each curve represents a distribution. The profiles are sorted by probability, so the profile with the highest probability appears on the far left. 68
4.2 Most stable settings of the ENS parameter τ across different levels of noisy observations on random games. Results for the stochastic observation model are on top, and the incomplete observation model on the bottom. Higher settings of τ correspond to predictions that are closer to random play. Observation noise is increasing left to right. 71
4.3 Most stable settings of the RDS parameter δ across different levels of noisy observations on random games. Results for the stochastic observation model are on top, and the incomplete observation model on the bottom. Higher settings of δ correspond to predictions that are closer to random play. Observation noise is increasing left to right. 72
4.4 Most stable settings of the LES parameter λ across different levels of noisy observations on random games. Results for the stochastic observation model are on top, and the incomplete observation model on the bottom. Lower settings of λ correspond to predictions that are closer to random play (opposite of ENS and RDS). Observation noise is increasing left to right. 73
4.5 **Uniform random games.** Comparison of four meta-strategies playing uniform random games with stochastic observations (top) and incomplete observations (bottom). 77

4.6 **Common interest games.** Comparison of four meta-strategies playing common interest games with stochastic observations (top) and incomplete observations (bottom). 78
4.7 **Constant sum games.** Comparison of four meta-strategies playing constant sum games with stochastic observations (top) and incomplete observations (bottom). 79
5.1 The game board for chaturanga. 86
5.2 Nine settings of the attack angle parameter (left), and equal weighting of differences (right). 92
5.3 Meta-strategy analysis for candidate exploration policies on uniform random games. 117
5.4 Meta-strategy analysis for candidate exploration policies on factored (5,2) games. 117
5.5 Meta-strategy analysis for candidate exploration policies on factored (2,5) games. 118
5.6 Meta-strategy analysis for candidate exploration policies on random LEG games. 118
5.7 Meta-strategy analysis for candidate exploration policies on supermodular games. 119

Abstract

Complex multi-agent systems often are not amenable to standard game-theoretic analysis. I study methods for strategic reasoning that scale to more complex interactions, drawing on computational and empirical techniques. Several recent studies have applied simulation to estimate game models, using a methodology known as empirical game-theoretic analysis. I report a successful application of this methodology to the Trading Agent Competition Supply Chain Management game. Game theory has previously played little—if any—role in analyzing this scenario, or others like it. In the rest of the thesis, I perform broader evaluations of empirical game analysis methods using a novel experimental framework.

I introduce *meta-games* to model situations where players make strategy choices based on estimated game models. Each player chooses a *meta-strategy*, which is a general method for strategy selection that can be applied to a class of games. These meta-strategies can be used to select strategies based on empirical models, such as an estimated payoff matrix. I investigate candidate meta-strategies experimentally, testing them across different classes of games and observation models to identify general performance patterns. For example, I show that the strategy choices made using a naïve equilibrium model quickly degrade in quality as observation noise is introduced.

I analyze three families of meta-strategies that predict distributions of play, each interpolating between uninformed and naïve equilibrium predictions using a single parameter. These strategy spaces improve on the naïve method, capturing (to some degree) the effects of observation uncertainty. Of these candidates, I identify logit equilibrium as the champion, supported by considerable evidence that its predictions generalize across many contexts.

I also evaluate exploration policies for directing game simulations on two tasks: equilibrium confirmation and strategy selection. Policies based on computing best responses are able to exploit a variety of structural properties to confirm equilibria with limited payoff evidence. A novel policy I propose—subgame best-response dynamics—improves previous methods for this task by confirming mixed equilibria in addition to pure equilibria. I apply meta-strategy analysis to show that these exploration policies can improve the strategy selections of logit equilibrium.

Chapter 1

Introduction to Games and Empirical Game Modeling

Humans act not in isolation, but in the context of a rich web of social interactions. They trade goods and services, cooperate to build large-scale infrastructure, engage in military conflicts, and compete in sporting events. The common thread in these examples is that the outcome may depend on the choices of many individuals, each with different objectives. To make good decisions under these circumstances, it is necessary to reason about the behavior of others.

Game theory originated in economics as a mathematical framework for modeling interactions among decision makers, beginning with the seminal work of von Neumann and Morgenstern (1944). The theory treats individual decision makers as players in a game, each of whom chooses a strategy (i.e., action) from among a set of alternatives. The value that each player receives depends on the strategies of all players in the game, not just their own action. To select the best strategy, players need to reason *strategically*, predicting what the other players are likely to do and responding accordingly. The literature on game theory is extensive, and provides rich formalisms for both representing and solving games. The popularity of game theory has grown rapidly since it was introduced, and it is now prevalent as an analytic tool in academic disciplines including economics, computer science, psychology, biology, and political science.

The field of artificial intelligence is broadly interested in designing autonomous agents that exhibit intelligent behaviors. Increasingly, autonomous agents must interact with both humans and other intelligent agents. The area of multi-agent systems research focuses on designing and analyzing systems with more than one decision maker. One objective is to design autonomous agents that are able to achieve high utility in the presence of others. This may require competitive or cooperative interactions with humans or other agents. Another broad objective is to design rules or institutions for agent interactions that achieve the objectives of the system designer. Designing such rules is often called *mechanism design*. For example, an auctioneer may want to set the rules of an auction in such a way as to maximize her own revenue. Both of these objectives—agent design and mechanism design—may require predicting what actions other agents will take.

There are many application areas for agent technologies that may involve interactions with either humans or other agents. In some examples, agents augment or replace humans: they act as intermediaries in electronic commerce (Mackie-Mason and Wellman, 2006), control robots (Asada et al., 1999), and provide adversaries for military training (Wray et al., 2005). Multi-agent systems are also an important tool for modeling human institutions and behavior, including applications in computational economics (Tesfatsion and Judd, 2006), and air traffic control (Tumer and Agogino, 2007).

It is natural to consider game theory as a paradigm for designing and evaluating multi-agent systems, and the approach has gained considerable popularity. However, multi-agent systems often pose significant challenges for game-theoretic methods. Game theory is usually concerned with finding exact solutions to relatively simple, stylized models of strategic interactions. Many multi-agent systems are complicated by large strategy spaces, many players, and uncertainty. These difficulties may render conventional game-theoretic solution methods infeasible, both analytically and computationally. Computer science offers methods with the potential for scaling game-theoretic analysis to much larger problems, including simulation, heuristic search, and structured representations.

This thesis contributes to a growing body of work that develops computational tools for strategic reasoning in very complex domains. My primary contributions advance the methodology of empirical game-theoretic analysis, which combines empirical methods (e.g., simulation) and game-theoretic principles to reason about very complex games. I illustrate the methodology with an application to a particularly challenging supply chain domain, adding to an existing body of work that uses similar approaches to study specific games. The literature contains examples of various methods for performing this analysis, but there has been little evaluation of which approaches are most effective. In this thesis, I move beyond specific applications to evaluate approaches for empirical game-theoretic analysis more generally.

A key insight is that empirical game-theoretic analysis can be modeled as a *meta-game* that captures the observation process, including the crucial aspect of payoff uncertainty. Based on this model, I develop criteria and an experimental framework for evaluating different analytic procedures for empirical game modeling. The framework is especially suited to broad evaluation across different conditions, such as different classes of games or methods of observing payoff information. I use this capability to test the strengths and weaknesses of the different approaches, identifying general performance patterns.

I study two broad questions using this framework. The first is how players should select strategies to play, given only noisy estimates of the payoffs in a game. I identify logit equilibrium as the champion for this task, and present considerable evidence that this method performs well in a variety of contexts. I also provide evidence that rational players have a preference for making broader predictions of opponent play as payoff uncertainty is increased. The second question I consider is how players should explore a large game, given limits on the amount of payoff information they are allowed to observe. I evaluate several exploration policies on two different analysis tasks, and show that they are able to exploit various types of structure to make good use of the available payoff information. A novel method I propose offers substantial improvements over previous approaches on an equilibrium identification task. A more detailed overview of the thesis appears in Section 1.4, after I have introduced some background material.

1.1 Game Definitions and Terminology

I begin by briefly introducing some necessary background material on the theory of games and relevant solution concepts. A thorough introduction to game theory can be found in many texts, such as Fudenberg and Tirole (1991) or Osborne (2004).

1.1.1 Normal-Form Games

The most basic representation of a game is the *normal form* (also known as *strategic form*). A game in the normal form specifies the players participating in the game, their possible strategies, and the payoff each player gets for each possible outcome.

Definition 1 *A* normal-form game *consists of a tuple $\langle I, S, U \rangle$ defining the set of players I of size n, sets of actions for each player $S = \{S_i\}$, and a payoff function for each player, $U = \{u_i(s)\}$. The payoff function gives the value that a player receives for each possible combination of strategy choices by all players, $s = [s_1, \ldots, s_n]$.*

The elements of each player's strategy set $s_i \in S_i$ are called *pure strategies*. Players may also use randomized *mixed strategies*, $\sigma_i \in \Sigma_i$, which specify a probability distribution over pure strategies. Pure strategies are also mixed strategies. A joint selection of strategies for all players in a game is a *strategy profile*, denoted $s = [s_1, \ldots, s_n]$ for pure strategies and $\sigma = [\sigma_1, \ldots, \sigma_n]$ for mixed strategies. The notation s_{-i} (respectively, σ_{-i}) refers to a profile of strategies for all players except player i. A *homogeneous profile* is one in which all players use the same strategy.[1]

Normal-form games are played once, and each player makes its strategy choice without knowledge of the other players' choices (i.e., simultaneously). The description of the game,

[1] This requires all players to have the same set of strategies, but does not require that players have identical payoffs for these strategies.

$< I, S, U >$, is generally assumed to be common knowledge among all players. Common knowledge is a strong condition that implies that all players know the game, know that all other players know the game, know that the other players know that they know the game, and so on in an infinite hierarchy of beliefs.

	L	R
U	3, 3	0, 4
D	4, 0	1, 1

Table 1.1 An example of a matrix representation of a normal-form game. This game is a Prisoner's Dilemma, one of the most famous examples in game theory.

Normal-form games are often represented using a payoff matrix, as shown in Table 1.1. This example has two players, a row and column player. The row player selects between actions U and D, and the column player selects between L and R. The first payoff listed in each cell is for the row player, and the second is for the column player. If the players choose strategy profile [U,R] the row player receives a payoff of 0 and the column player receives a payoff of 4.

An important special case is *symmetric games*, in which all players have the same strategy choices and identical payoff functions. That is, $S_1 = \cdots = S_n$, and $u_i(s_i, s_{-i}) = u_j(s_j, s_{-j})$ when $s_i = s_j$ for all pairs of players $i, j \in I$. There are no distinct player roles in a symmetric game, and the payoffs depend only on the strategies selected and not the identity of the players. An important advantage of symmetric games is that they have far fewer distinct strategy profiles than general games. This can be exploited to reduce the size of the representation and design more efficient solution algorithms (Papadimitriou and Roughgarden, 2005).

1.1.2 Bayesian Games

The normal-form game model makes the strong assumption that the game is common knowledge among all players. Of course, in many interesting situations, players are uncertain about aspects of the game and may not share the same information. Game of incomplete information, or *Bayesian games*, were introduced by Harsanyi (1967–1968) to model uncertainty in games. In particular, players may have private information about the payoffs in the game, and may be uncertain about what other players know about the payoffs.[2]

Bayesian games introduce the notion of a player's *type*, which encapsulates all of the private information a player has about the payoffs. The payoff functions in a Bayesian game depend on both the strategy selections (as in normal-form games) and the types of all players. In addition, each player may condition its strategy choice on its type. The type of each player is determined randomly at the beginning of the game by a special player known as "Nature." Nature has no interest in the game, but selects the type of each player from a probability distribution and reveals it only to that player. The type distributions are generally assumed to be common knowledge among the players. This knowledge can be used by players to compute Bayesian beliefs about other players' types and the payoffs in the game.

Poker is an intuitive example of a Bayesian game. Each player in the game has a type determined by its hand of cards, which is drawn randomly from a shuffled deck (represented by the move of Nature in the abstract game). Players bet based on their own hand, and the beliefs they hold about other players' hands. At the end of the game, payoffs are determined by a combination of each players' actions (bets) and the type of each player (its hand).

1.2 Solution Concepts for Games

Analyzing a game form to predict the outcome or to select a good strategy to play is a surprisingly subtle task. The game theory literature is rife with solution concepts, each with unique properties and motivations. Here I highlight only a few of the most important concepts, and those which play a role in later chapters.

[2] Harsanyi argued in the original paper that uncertainty can be restricted to uncertainty about payoffs without loss of generality, since other forms of uncertainty can be represented as payoff uncertainty.

	L	M	R
U	6, 2	11, 4	5, 5
D	7, 4	10, 10	4, 2

Table 1.2 Example game 2, in normal form.

1.2.1 Dominance

One goal of analyzing a game is to identify the best strategy for a given player. In general, which strategy is best depends on the choices made by the other players, so this is not a simple optimization problem. However, in some cases a strategy may have a higher payoff than another strategy against all possible opponent strategy choices.

Definition 2 *A pure strategy s_i is* dominated *if there exists a strategy σ_i such that $u_i(\sigma_i, s_{-i}) \geq u_i(s_i, s_{-i})$ for all profiles of opponent strategies $s_{-i} \in S_{-i}$. If the inequality is strict, σ_i* strictly dominates *s_i. A* dominant strategy *is a pure strategy that dominates all of a player's other pure strategies.*

In example game 2 (shown in Table 1.2), the column player's strategy L is dominated by strategy M. Dominant strategies do not always exist, but they offer a compelling solution criterion when they do. Even when there is no single dominant strategy, the idea of dominance is often useful for pruning the space of strategies under consideration.

1.2.2 Nash Equilibrium

When a player does not have a dominant strategy, the next best alternative is to play a strategy that gives a high payoff against strategies other players are likely to choose. Reasoning about the other players to predict their choices becomes a key part of the analysis. Many solution concepts focus on identifying profiles of strategies that are in *equilibrium* with each other. These profiles are often considered more likely outcomes because they have desirable stability and consistency properties. Before delving into a discussion of equilibrium I introduce the useful notions of *deviation* and *best-response*. Unless otherwise noted, a "deviation" always refers to a unilateral deviation by a single player.

Definition 3 *A player may* deviate *from any strategy profile by changing to any pure strategy in its strategy set. The set of profiles resulting from unilateral deviations from the profile* $\sigma = [\sigma_1 \ldots, \sigma_i, \ldots, \sigma_n]$ *by player i is* $\mathscr{D}_i(\sigma) = \{[\sigma_1, \ldots, s'_i, \ldots, \sigma_n] : s'_i \in S_i\}$.

Definition 4 *A* best response *for player i to a profile of strategies for the remaining players* σ_{-i} *is a pure strategy with maximal expected payoff against the profile. There may be more than one best response, but all have the same expected payoff. The set of best responses is given by* $\mathscr{B}_i(\sigma_{-i}) = \{s'_i : u_i(s'_i, \sigma_{-i}) \geq u_i(s_i, \sigma_{-i}) \forall s_i \in S_i\}$.

A profile where every player simultaneously plays a best response (i.e., there are no beneficial deviations) is a Nash equilibrium (Nash, 1951). This is one of the most influential concepts in game theory, and many other solution concepts are refinements of this idea.

Definition 5 *A* Nash equilibrium *is a strategy profile* σ *with* $u_i(\sigma_i, \sigma_{-i}) \geq u_i(s'_i, \sigma_{-i})$ *for all players i and pure strategies* $s'_i \in S_i$*. If all strategies in the profile* σ *are pure strategies, this is called a pure-strategy Nash equilibrium (PSNE).*

In example game 2 (Table 1.2), the profile [U,R] is a Nash equilibrium, as both deviations result in a lower payoffs. Nash equilibrium has several appealing properties. Unlike dominant strategies, Nash equilibria are guaranteed to exist in most cases of interest. In normal-form games there will always be at least one mixed-strategy equilibria, though pure-strategy equilibria may not exist. Nash equilibria are consistent with rational behavior by the players in the sense that no player is making an obvious mistake (by failing to optimize or having inaccurate beliefs). In other words, no player would *regret* their decision after the outcome is revealed. The notion of regret leads to a natural generalization of Nash equilibrium.

Definition 6 *The* regret *for a pure strategy profile, denoted $\varepsilon(s)$, is the maximum benefit any player could get by deviating to an* alternative *pure strategy:* $\varepsilon(s) = \max_{i \in I, s'_i \in \{S_i \setminus s_i\}}[u_i(s'_i, s-i) - u_i(s)]$. *The regret for a mixed strategy profile containing at least one strategy that is not pure, denoted $\varepsilon(\sigma)$, is the maximum benefit any player could achieve by deviating to any pure strategy:* $\varepsilon(\sigma) = \max_{i \in I, s'_i \in S_i}[u_i(s'_i, \sigma_{-i}) - u_i(\sigma)]$. *Note that this definition applies to all strategy profiles, not just equilibrium profiles.*

This is a somewhat non-standard definition for regret in the case of pure-strategy profiles, where I exclude the original strategy from the calculation. The consequence of this is that negative values of regret are permitted for these cases, which I use in several places to make distinctions among pure Nash equilibria. A Nash equilibrium has no regret, by definition ($\varepsilon(\sigma) \leq 0$). An approximate Nash equilibrium is a profile σ with low regret; we say that a profile is an ε-Nash equilibrium if it has regret $\varepsilon(\sigma)$.

Despite its appeal, there are a number of well-known theoretical and practical difficulties associated with Nash equilibrium. Nash equilibrium does not necessarily define a unique solution, because games may have a multiplicity of equilibrium profiles. Many equilibrium refinements have been proposed to select among the set of Nash equilibria by applying other considerations. A few examples are trembling-hand perfection (Selten, 1975), proper equilibrium (Myerson, 1978), sequential equilibrium (Kreps and Wilson, 1982), and evolutionary stability (Smith, 1982; Taylor and Jonker, 1978).

Computing Nash equilibrium also presents computational difficulties. Known general solution algorithms (many of which are implemented in the Gambit software package (McKelvey et al., 2006)) do not scale well to large games.[3] Theoretical complexity results indicate that efficient (polynomial time) algorithms are unlikely to be found (Daskalakis et al., 2006).

There are persistent questions about the value of Nash equilibrium as a predictive model. Results from experimental studies show that in many circumstances, humans do not play as predicted by equilibrium models (Goeree and Holt, 2001; Erev et al., 2002; Mailath, 1998;

[3] As a rough guideline, games with more than a few players and/or hundreds of actions are problematic.

Erev and Roth, 1998; Sarin and Vahid, 2001). This has led to a number of proposed solution methods based on weaker rationality assumptions, including rationalizability (Pearce, 1984; Bernheim, 1984) and forms of worst-case analysis (Aghassi and Bertsimas, 2006; Tennenholtz, 2002). I discuss several other approaches motivated by this observation in later chapters, including noisy equilibrium models and dynamic learning models.

1.3 Empirical Models of Games

Game-theoretic analysis typically begins with a game model specifying the players, strategies, and payoffs. The model may serve different purposes, depending on the goals of the analyst. One common objective is to derive general insights into strategic reasoning, ideally with broad applications to many situations. Simple stylized games are often well-suited to this endeavor, since they may provide sharp examples of interesting phenomena. Such games are typically abstract, and not intended to precisely model any specific situation. For example, the Prisoner's Dilemma is a widely studied game that provides a simple example of how cooperative behavior can be obstructed by individual incentives. It is often introduced using an intuitive narrative about two imaginary prisoners offered a deal during interrogation by the law enforcement authorities. However, the point of analyzing the model is to better understand the general phenomenon of cooperation in games, rather than to offer guidance to the two imaginary prisoners.

An analyst may also have interest in a specific game, whether to participate, offer advice, or for some other reason. For this purpose, the main concern is whether the model is an accurate representation of the strategic situation. Specifying a suitable model of a complex multi-agent system such as poker (Billings et al., 2002), RoboCup (Asada et al., 1999), or the Trading Agent Competition games (Wellman et al., 2001; Arunachalam and Sadeh, 2005) is not trivial. It is telling that no exact game-theoretic solution (e.g., Nash equilibrium) has been reported for the full version of any of these games, despite substantial research efforts to develop strategies for these games. In the next chapter I discuss at length the difficulties presented by the Trading Agent Competition Supply Chain Management game.

One fundamental problem is that evaluating the payoffs for these games is costly because they are not given in a direct mapping of strategy profiles to payoffs. Consider the RoboCup domain, in which teams of robotic agents (or simulated robots) compete in variants of soccer. This game is defined by a set of rules that specify the legal moves and the objective—score more goals than the opponent. Each instance of the game is a sequence of legal moves, with some random elements (e.g., the direction the ball bounces). For any two teams of robots, it is not possible to say with certainty what the result of a game would be without actually playing the game.[4] Playing games to estimate the payoffs requires both time and resources, even though the game is well-defined. Coupled with the very large number of possible strategies, the cost of determining payoffs makes it infeasible to derive a complete and exact model of the game for analysis.

One approach for analyzing these games is to rely on the analyst to define an abstract game that is tractable and encapsulates the relevant strategic aspects of the game. However, it is not clear that the analyst's intuition will yield an accurate model of the game. There is also little hope for assessing the quality of the model if the process used to create the model is left undefined.

An alternative approach is to use data (e.g., from simulations) to estimate a model of the game. Figure 1.1 outlines a high-level methodology for using empirical methods to build and analyze game models, which I refer to as empirical game-theoretic analysis. The basic stages in the analysis are:

1. Select an initial set of candidate strategies for each player
2. Use simulation to estimate the payoffs in the game
3. Apply game-theoretic methods to analyze the estimated game
4. Refine the analysis, either by running additional simulations or adding new candidate strategies (optional)

I describe each stage briefly here, but provide describe them in more detail in the context

[4] If it were, the tournament competitions would be much less interesting!

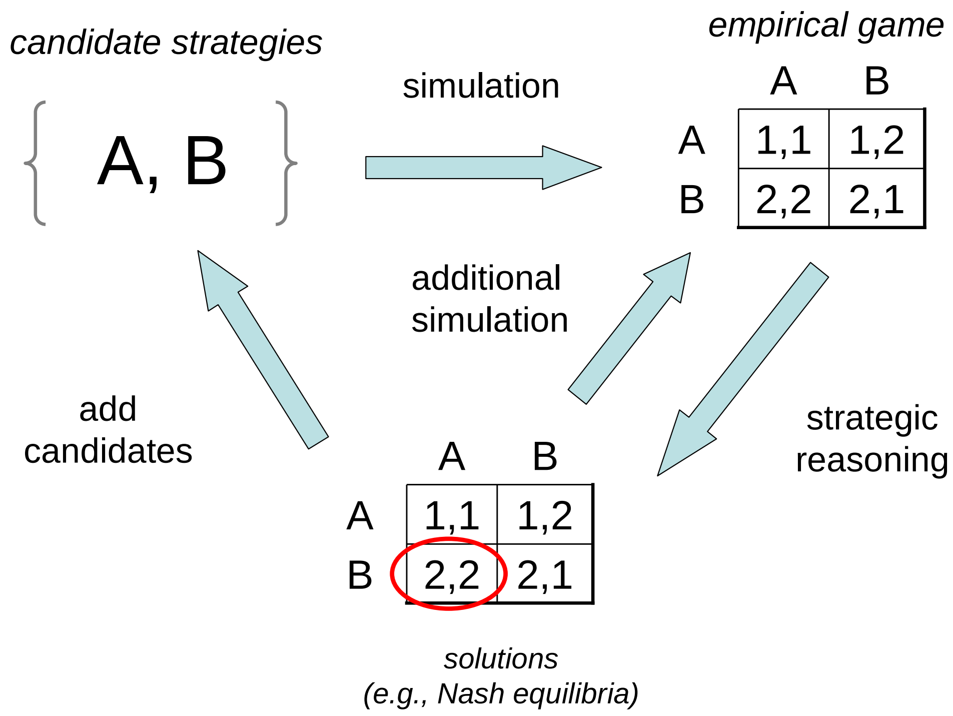


Figure 1.1 Steps in generating and analyzing an empirical game model.

of an application in the next chapter. In the first stage, the analyst chooses a set of candidate strategies, which may be relatively small compared to the full strategy space. This restriction is often necessary if the game is very large to keep the costs of estimating and analyzing the empirical game manageable. Additional candidates may be added as the analysis progresses. In the second stage, evidence is gathered to estimate the payoffs for profiles of candidate strategies. For example, instances of the game may be simulated using the candidate strategies to sample the payoffs for a profile. These payoff estimates can be used as the basis for game-theoretic analysis, such as finding a Nash equilibrium of the estimated game.[5] Based on the results of this analysis, the analyst may seek new evidence or consider additional

[5] Depending on the type of analysis desired, it may not be necessary to have estimates for all of the strategy profiles.

candidate strategies.

An early version of this empirical approach was introduced by Walsh et al. (2002), who computed what they called a heuristic payoff table to study strategies for two games, a continuous double auction and an automated dynamics pricing game. They studied several candidate strategies, based on methods found in the literature. A series of papers by MacKie-Mason, Osepayshvili, Reeves, and Wellman further develop this methodology, primarily in a market-based scheduling domain (MacKie-Mason et al., 2004; Osepayshvili et al., 2005). Reeves (2005) presents the most complete description of the methodology, as well as applications to several additional domains. I describe some additional studies that apply similar methods in Section 2.4. This literature and the application I present in Chapter 2 demonstrate the value of this methodology for analyzing a variety of complex strategic phenomena. However, there are many open questions about how best to implement the various task in empirical game-theoretic analysis, including:

- How should the candidates strategies be selected?
- When should additional candidate strategies be added?
- Which profiles should be sampled to best estimate payoffs in the empirical game?
- What solution methods should be used on estimated games?
- What is the degree of confidence in the solution?

While the literature contains examples of ways to answer each of these questions, there is at best sparse evidence to discern which methods are most effective. The answers may be complex, and depend on a variety of factors including the properties of the game and the methods available for gathering information about the game. One of the contributions of this thesis is a principled methodology and criteria for evaluating these methods, leading to new insights into the effectiveness of different techniques in a various contexts. I focus particularly on the questions of which solution methods should be applied to estimated games (Chapter 4), and which profiles should be sampled to support different analysis tasks (Chapter 5).

1.4 Thesis Overview

Many of the results presented in the sequel derive from and extend published papers, in some cases with several co-authors. I overview the main topics of each chapter, and highlight the relevant publications. Chapter 2 presents an application of empirical game-theoretic analysis to the Trading Agent Competition Supply Chain Management game. This serves several purposes, illustrating some challenges of strategic reasoning in complex multi-agent systems, introducing the empirical methodology, and providing evidence that the methodology can produce relevant insights in very complex domains. The material in this chapter is based on work presented by Jordan et al. (2007), which builds on our earlier work applying similar methods to reason about various strategic aspects of the game (Wellman et al., 2005a; Vorobeychik et al., 2006). In addition, many individuals have contributed to the design of our agent for this game over several years of tournament competition (Kiekintveld et al., 2004, 2006a, 2008).

The broad goal of the remaining chapters is to develop and evaluate specific methods for empirical game-theoretic analysis. One of the most salient concerns of relying on empirical game models as a basis for decision making is that they are typically noisy and/or incomplete representations of the game. In Chapter 3 I present a meta-game that models strategy selection using noisy empirical games (Kiekintveld and Wellman, 2008). Based on this model, I develop an experimental framework for evaluating strategy selection methods (i.e., meta-strategies). This approach is particularly useful for assessing the performance of meta-strategies across varying conditions, such as different magnitudes of noise. I conclude the chapter by demonstrating this framework to analyze strategy selection based on naïve equilibrium analysis. The consequences of noise in the observed game are very apparent in this simple experiment.

Chapter 4 builds on material from Kiekintveld and Wellman (2008). I apply the framework from Chapter 3 to study three families of parameterized meta-strategies. Each meta-strategy forms a distributional prediction of play, predicting the probability that opponents will select each possible combination of pure strategies. I study meta-strategies that make a range of predictions defined by varying a single parameter in each case, the effect of which interpolates the prediction between an uninformed prediction and a Nash equilibrium prediction. I test the performance of these algorithms experimentally across several classes of games, varying the payoff information that the meta-strategies have available for making predictions. My first experiment shows a systematic relationship between the gross level of payoff information available and the precision of the best predictions of opponent play. The second provides a comprehensive comparison of the three meta-strategies, and identifies one meta-strategy as the current champion for this task.

Chapter 5 addresses the problem of deciding which profiles of a game to explore in cases where it is not possible to observe the payoffs for all profiles. The germination of this work was a study of chaturanga (4-player chess) using empirical game-theoretic methods (Kiekintveld et al., 2006b). I describe a parameterized strategy space for chaturanga and examine some evidence for structure in this strategy space. The remainder of the chapter evaluates several families of exploration policies on game classes with known structural properties. I test these policies for both an equilibrium confirmation task and the strategy selection task, applying the meta-strategy analysis methods of Chapter 3 to evaluate the latter.

Chapter 2

Empirical Game-Theoretic Analysis Applied to the TAC Supply-Chain Game

Research in multi-agent systems often considers complex interactions among sophisticated agents. Analyzing these systems is challenging for various reasons, including large numbers of players, large strategy spaces, and uncertainty. Finding exact game-theoretic solutions (e.g., equilibria) is infeasible or impractical in many instances. However, it is possible to make progress in these domains by applying experimental evaluation. The paradigm of empirical game-theoretic analysis provides a foundation for such experimentation, drawing on the insights and tools of game theory.

The Trading Agent Competition is an annual event that challenges researchers to design strategies for market games. I focus here on the Supply Chain Management game. Despite a great deal of investigation by researchers since the game was introduced in 2003, there is no known optimal strategy or equilibrium of the game, and no exact solution of this form appears imminent. I illustrate the methodology of empirical game-theoretic analysis using this domain, demonstrating the viability of this approach for scaling game-theoretic analysis to domains that are not amenable to standard analytic approaches.

2.1 The TAC Supply Chain Management Game

The Trading Agent Competition provides common domains and infrastructure for multi-agent systems research.[1] The competition routinely attracts attention from a diverse group of researchers spanning academia and industry, with strong international participation. The first competition in 2000 featured a travel shopping domain (Wellman et al., 2001). The supply chain management game (TAC SCM) was added in 2003 (Arunachalam and Sadeh, 2005; Eriksson et al., 2006). I provide a brief description here, sufficient to understand the material in this chapter. The interested reader is encouraged to refer to Collins et al. (2006) for the canonical game specification.

2.1.1 Game Description

In the SCM game six fully autonomous manufacturing agents compete to maximize their profits in a simulated supply chain for personal computers (PCs). They manage component procurement, PC sales, and factory operations for 220 simulation days, each lasting 15 seconds. There are 16 types of PCs, defined by compatible combinations of 10 different components including motherboards, processors, memory, and hard drives. Each agent is endowed with an identical factory capable of producing all types of PC, but with limited overall production capacity. Inventory is subject to storage costs, and there is an interest rate for capital. At the end of the game, agents are evaluated base on their total profit. A visual overview of the supply chain configuration is given in Figure 2.1.

Agents must negotiate in markets to purchase components from suppliers and sell their finished PCs to customers. Negotiations in both the component and PC markets take place using a request-for-quote (RFQ) mechanism. Suppliers and customers are independent agents, with behaviors defined in the game specification and implemented by the server. Suppliers have a limited production capacity that changes each simulation day according to

[1] See http://www.sics.se/tac and http://tradingagents.org for background information about the competition, game specifications, tournament results, and other related information.

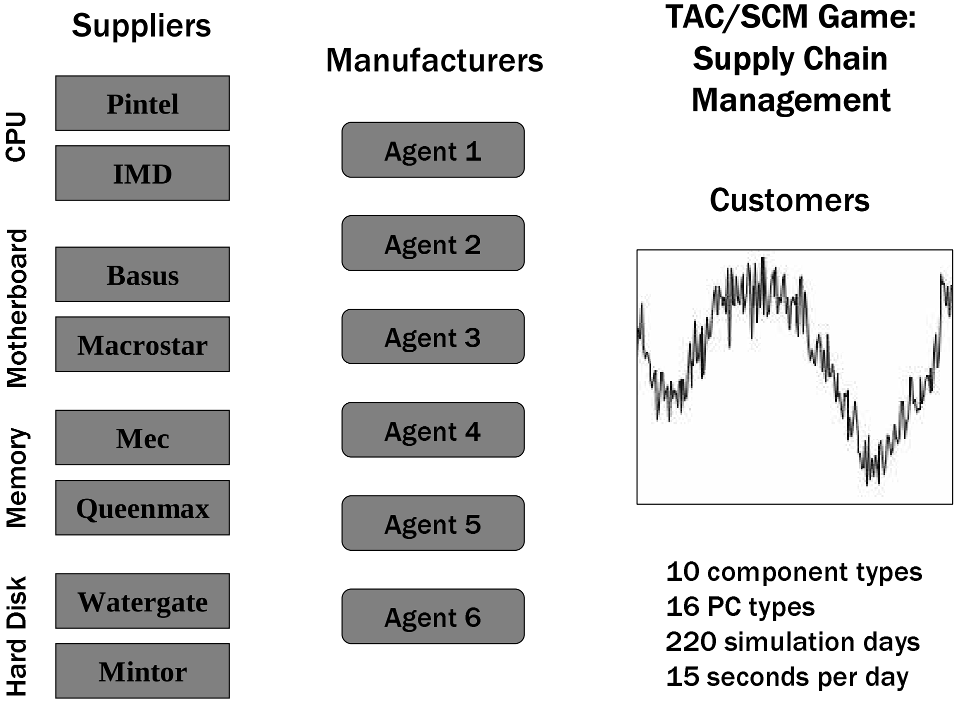


Figure 2.1 Diagram overviewing the configuration of the TAC SCM game.

a mean-reverting random walk. They respond to requests with offers if they have available capacity, pricing offers roughly based on the fraction of their capacity they have available. As demand for components increases, suppliers commit more capacity to orders, and prices generally rise. A set of customer requests is generated each day by the game server, with the number of requests reflecting the demand level. Demand varies each day according to a stochastic process with a trend parameter. The customer requests are treated as simultaneous first-price sealed-bid auctions.

Manufacturers must make decisions based on limited information about the state of the other agents in the supply chain, including the underlying supplier capacities and customer demand. The decisions of other manufacturers play an integral role in determining market conditions, so uncertainty about the strategies employed by the other agents is a key aspect of the game.

2.1.2 Strategic Interactions in TAC SCM

The performance of any manufacturing agent in the TAC SCM game depends on the choices of the other agents, and this interdependence has played a pivotal role in the competition. One dramatic example is the case of early component procurement during the first two years of the TAC SCM competition. There was a very powerful incentive in first version of the game to place component orders before other agents, since early orders cause the price for later orders to increase. Agents gradually migrated towards very aggressive procurement strategies during the early tournament rounds, and eventually the majority of component purchases were occurring on the very first day of the game. However, this had a mutually destructive effect on agent profits, because these purchases were made with essentially no information about what the customer demand conditions were going to be.

We conducted an empirical game-theoretic analysis of this phenomenon using variations of our own agent, Deep Maize (Wellman et al., 2005a). Our experiments verified that the aggressive policies observed were rational (in a limited sense), but were mutually destructive to manufacturing profits. However, a "preemptive" strategy introduced by Deep Maize in the final round neutralized the aggressive behavior and resulted in a new equilibrium in which profits for all manufacturers (not only Deep Maize) were higher.

The emphasis on very early procurement was considered a design flaw by the TAC SCM community, and revised rules were introduced to mitigate the issue. However, the 2004 changes that focused on introducing storage costs were unsuccessful in effecting the desired change in agent behavior (Kiekintveld et al., 2006c). We conducted further experiments on possible settings of this storage cost parameter using empirical mechanism design. This analysis showed that no reasonable setting for the storage cost would have been likely to reduce incentives for early procurement to desirable levels (Vorobeychik et al., 2006). A further redesign the following year overhauled the supplier model and substantially reduced incentives for the extreme early procurement policies. This anecdote demonstrates both the inherent strategic nature of the TAC SCM game, and the difficulty of anticipating likely behaviors in the game (even for the designers of the game rules).

2.1.3 Challenges for Strategic Analysis

The fundamental challenges for analyzing strategic choices in TAC SCM are common to most real domains and applications of multi-agent systems. A strategy for playing TAC SCM (i.e., a manufacturing agent) must specify choices for a large number of distinct decisions, conditional on several types of observations. The strategy space defined by the set of all possible choices for these decisions is immense. Consider the number of ways to bid on the set of customer requests for a *single* simulation day. During periods of low customer demand, there are roughly 100 customer requests each day, and agents may bid any integer value up to the reserve price—roughly 2000 distinct values. There are 2000^{100} distinct sets of bids that the agent could place on the set of requests. In other words, there are more than 10^{300} ways for the agent to bid on the customer requests for a single day, even during periods of low demand. A complete agent strategy must specify customer bids for the entire game, as well as strategies for procurement and manufacturing activities.Adding these additional decisions can only increase the size of the strategy space, so the number of complete agent strategies is extremely large.[2]

Enumerating the strategy space for TAC SCM is clearly infeasible, and the strategies themselves are too complex to represent using tabular mappings from observations to actions. Instead, strategies are represented in compact form as algorithms, i.e., software agents that specify how to compute decisions. Tournament agents often comprise thousands of lines of computer code. An important implication of this for analyzing the strategy space is that there is a computational cost for executing the strategy.

[2] It is difficult to provide a precise measure of the number of complete strategies, due to stochastic elements in the game, and because some decision variables are not explicitly bounded.

The rules for TAC SCM are also a compact representation of the game. The specification comprises roughly 20 pages of text, which is defines the behavior of the game server implementation. To play an instance of the game, agents connect to the game server and communicate their own decisions for each simulation day, while the game server responds as described in the game specification. For any given set of agents, the game can be simulated to determine the sequence of messages between the server and the agents, and the final profits. The game has stochastic elements, so these payoffs represent only a sample from the distribution of payoffs for these agents. To achieve an accurate estimate of the expected payoffs, many simulations are necessary. The key point is that there is no way to discover the payoffs for a set of agents without actually running simulations; there is no simple formula for computing the payoffs for a set of agents. The computational costs associated with running simulations limits the amount of payoff information it is possible to gather. As a consequence, any available model of the game is necessarily a noisy and incomplete representation of the true game.

2.2 Empirical Game Modeling for TAC SCM

Like many other games, TAC SCM cannot be solved directly using analytic methods. To evaluate strategy choices, we must rely on experimental evidence. The TAC tournament is one important source of evidence for evaluating the quality of agent strategies. Tournaments offer a number of advantages, but one of the most important is the realistic strategic environment. There is a real potential to encounter unanticipated strategies in a tournament where agents are introduced by a diverse pool of participants.

However, tournament evaluation is limited in some important ways, both theoretical and practical. Tournament play consists of a limited number of game instances, and not all tournament results are statistically significant. This also limits the number of strategy variations designers can test in tournament conditions. There are many possible ways of matching agents (i.e., strategy profiles) that are never explored in a tournament setting. To some degree, this limits the conclusions that may be drawn from tournament results. For instance, it is generally not possible to identify equilibrium strategies using a tournament (even for the restricted set of tournament agents). Other issues included the lack of complete documentation of the participating agents, and the possibility that tournaments distort the incentives of the game away from strict profit maximization to focus on the value of a tournament ranking.

Many published accounts of agents for TAC supplement the tournament results with controlled experiments designed to test specific design choices (He et al., 2006; Pardoe and Stone, 2006; Benisch et al., 2006). Controlled experiments are also used extensively for testing potential strategies and tuning parameters leading up to the tournament. Such experiments often fix a set of opponents to provide competition and vary one or more aspects of a particular agent's strategy. Though it is possible to use the "dummy" agents supplied by the game server, using agents from previous competitions provides an environment that is closer to tournament competition. Many teams voluntarily release binary versions of their agents after each tournament, and the binaries are collected in a public agent repository.[3]

We follow the approach outlined in Section 1.3 to create empirical game models for exploring strategic interactions in TAC SCM. This source of experimental evidence complements the tournament results, and offers a more principled means for experimental design than simply selecting arbitrary opponents. The discussion that follows is not intended to argue for specific approaches for building or analyzing an empirical game model, but to offer a concrete example of how the general framework can be applied to achieve insights into behaviors in very complex domains.

[3] The repository is hosted at http://www.sics.se/tac/showagents.php.

2.2.1 Reducing the Game

We begin by restricting the game in several ways to reduce the size of the game and make estimating the payoffs more manageable. First, we exploit the fact that all manufacturers in TAC SCM are identical and consider the symmetric game. In addition to reducing the number of profiles that must be simulated, we can reduce sampling noise by averaging the payoffs for all identical strategies in a game instance. We also reduce the number of players in the game from 6 to 3 using the hierarchical player reduction technique developed by Wellman et al. (2005b). Each player in the reduced 3-player game effectively selects a strategy for a pair of players in the original game, so the reduced profile $[1,2,3]$ maps to $[1,1,2,2,3,3]$ in the original game. Game simulations still use six manufacturing agents, but there are always at least two identical agents. This reduction empirically yields good approximations with significant computational savings (Wellman et al., 2005b; Reeves, 2005).

Even using symmetry and player reduction, the strategy space is still far too large to explore exhaustively, so we focus on small sets of candidate strategies. While candidates could be chosen for a variety of reasons, it is natural to consider tournament agents available in the repository, as well as parameterized variations of existing agents as a starting point. Walsh et al. (2002) coined the term *heuristic strategy analysis* to describe their analysis of a restricted space of strategies. Armantier et al. (2007) use restricted strategy spaces as the basis for a method of approximating Bayes-Nash equilibrium. They refer to the game where players select only the restricted strategies as the *constrained game*, and equilibrium of the constrained game as *constrained equilibria*. Under certain conditions, the equilibria of a sequence of constrained games will converge to equilibria of the full game. In the sequel, I will use the terminology of constrained games and constrained equilibria to refer to games with restricted strategy spaces.

It is natural to ask what (if anything) can be inferred from analysis of arbitrarily constrained games. After all, any analysis that considers only a subset of the strategies runs the risk of ignoring promising alternatives. First, observe that the regret for a strategy profile in a constrained game is a *lower bound* on the regret for the same strategy profile in the full game. This immediately implies that any profile that is *not* an equilibrium of the constrained game cannot be an equilibrium of the full game. A constrained equilibrium may be an equilibrium of the full game, but this can be confirmed only by checking the deviations to all remaining strategies in the full game. There is a degree of supporting evidence for constrained equilibria, as a result of the deviations tested within the constrained game. If we suppose (conservatively) that all deviations from a given profile are equally likely to be beneficial, a constrained equilibrium is more likely to be an equilibrium of the full game than any profile that is not part of the constrained game.[4]

2.2.2 Estimating the Payoff Matrix

We use simulation to estimate the payoffs in the constrained game for the sets of candidate strategies. Each simulation requires an instance of the game server and six agents to control the manufacturers. To ensure fair allocation of computational resources, we allocate one CPU core for each agent and one for the game server. Each game instance runs for one hour, so each simulation requires approximately 7 hours of CPU time. The game server produces a log file for each game, containing detailed information about all messages and transactions in the game. After screening for failed simulations[5] we extract scores for each agent from the game logs. We use the statistical method of control variates (Ross, 2002) to reduce the variance in the scores, effectively adjusting scores to account for the level of customer demand observed in each game instance. This method was initially applied in Wellman et al. (2005a). Another of variance reduction based on modifications to the game

[4] This reasoning is even stronger if the strategies in the constrained game are "better" than the average strategy in the full game, a reasonable assumption in many cases. For instance, it would be very surprising for a randomly-generated TAC SCM strategy to be a beneficial deviation from a profile of tournament agents.

[5] Games are scratched if any agent did not communicate with the server for six or more days. This screens for problems such as communication failures and crashed agents, while allowing some normal behaviors where agents do not communicate. Failure rates are generally very low in our computing environment.

server has recently been developed by Sodomka et al. (2007).

Estimating a complete payoff matrix for even a modest number TAC SCM strategies is very expensive. Even using symmetry and player reduction, collecting 30 samples for each profile in a 3-player version of the game with 5 pure strategies requires over 7000 CPU-hours. Fortunately, it is simple to parallelize the process of simulating a game matrix. This allows us to harness substantial resources provided by a computing cluster at the University of Michigan. We manage simulations using a centralized server to submit and track jobs, gather game logs, and run post-processing tasks that extract scores and build payoff matrices. The current implementation supports only relatively simplistic methods for determining which simulations to run, though more intelligent search methods such as those studied in Chapter 5 could potentially be used to improve this process. On the cluster, a set of scripts automatically configure the game server and agent strategies for each game instance requested. To date we have collected over 25,000 games for TAC SCM using this infrastructure.

There is one important caveat to bear in mind when comparing our simulation results to the tournament results. Tournament agents can maintain state between game instances, so they can adapt within a round of tournament games. Several agents take advantage of this opportunity, including the top-scoring agent from both 2005 and 2006 tournaments, TacTex (Pardoe and Stone, 2006, 2008). Our simulations are isolated game instances, with no concept of a round of games. Therefore, the strategies we consider can adapt only within a game instance.

2.3 Game-Theoretic Analysis of TAC SCM

Once we have created an empirical model of the game using simulation evidence, we analyze the model using game-theoretic principles. In this section I present results for tournament agents from the 2005 and 2006 competitions, all of which are publicly available in the agent repository. I highlight several methods that we have found useful for analyzing the data generated by our experiments, including ranking strategies based on performance in an equilibrium context and a graphical depiction of the profile stabilities and best-response correspondence. These results supplement the tournament results in several ways, demonstrating the value of the empirical game-theoretic paradigm.

2.3.1 Analysis of SCM 2006 Agents

The agents that competed in the 2006 TAC SCM finals are listed in Table 2.1, along with the scores they achieved in each round of the TAC tournament. Binaries for five of these agents were released to the TAC SCM agent repository:

- TacTex (Tx) (Pardoe and Stone, 2008)
- PhantAgent (Ph) (Stan et al., 2006)
- Deep Maize (Ds and Df) (Kiekintveld et al., 2006a)
- Maxon (Ms and Mf)
- MinneTAC (Mt) (Collins et al., 2007)

For Deep Maize and Maxon, the versions that played in both the semi-final and final round are available.[6] The MinneTAC agent available played in the semi-final round; according to the release statement, changes introduced for the final version resulted in significantly worse performance. In this section we present data for five of the seven available agents, including both versions of Deep Maize but excluding both Maxon agents.[7] The full symmetric game (reduced to 3 players) for these five agents has 35 unique profiles. Our empirical estimate for this payoff matrix is based on over 1100 simulated games, with a minimum of 15 samples for a profile, and typically at least 30.

Figure 2.2 provides a concise summary of the game data in visual form, depicting both the regret for each pure strategy profile and the best deviation from each profile. Each node in the graph represents a profile of three agents, each of which controls two manufacturers

[6] A significant change was made to Deep Maize's procurement policy for the start of the game between the semi-final and final round.

[7] Maxon was the last agent to be released, and we do not have simulation data for all of the necessary profiles to include the two variations of this agent in this part of the analysis.

Agent	Affiliation	Final	Semi-Final	Quarter-Final	Seeding
TacTex	U Texas	5.85	7.55 [2]	7.48 [B]	13.73
PhantAgent	Politech U Bucharest	4.14	5.71 [2]	17.37 [C]	12.56
DeepMaize	U Michigan	3.58	6.46 [1]	9.61 [A]	16.60
Maxon	Xonar Inc.	1.75	4.08 [1]	17.74[D]	10.63
Botticelli	Brown U	0.48	1.94 [1]	0.83 [A]	4.21
MinneTAC	U Minnesota	–2.70	2.06 [2]	13.45 [C]	9.59

Table 2.1 TAC SCM-06 finalists, with average scores ($M) from seeding through final rounds (semi-final and quarter-final groups in brackets).

in the full game. The placement of the nodes is based on the regret level for the profile, $\varepsilon(s)$. The concentric circles contain nodes within a range of ε values, with lower values of ε closer to the center of the graph. The profiles in the innermost circle are the most stable—either Nash equilibria or close approximations. The edges between the nodes represent the best deviation from each profile (the best response with the greatest benefit, over all players). A solid arrow means that the benefit to deviating is significantly greater than zero for $p \leq 0.05$.

In the graph we can see that there is a single Nash equilibrium in pure strategies, [Ds,Ph,Tx], which has no outgoing edges signifying beneficial deviations. There is also an approximate equilibrium [Ds,Ds,Tx] with a small (statistically insignificant) benefit of $0.09M$ for deviating to the equilibrium profile. We applied replicator dynamics (Taylor and Jonker, 1978; Reeves, 2005) to search for mixed-strategy equilibria, starting from mixtures generated at uniform random. In all cases the dynamics converged to the symmetric Nash equilibrium mixture of TacTex, PhantAgent, and Deep Maize SF, with probabilities [0.254, 0.188, 0.558]. This is also the equilibrium identified in the limit by the logit equilibrium solver in Gambit (McKelvey et al., 2006; Turocy, 2005).

Table 2.2 presents several additional statistics about the deviations in the 2006 game. *Percent positive deviations* is the fraction of possible deviations to the agent that result in a net benefit. *Best Deviation* is the number of instances where deviating to the agent was the most beneficial deviation. *Mean Deviation* and *std. error* reflect the average benefit ($M) for deviating to this agent, which may be negative. Deviations to TacTex, PhantAgent, and

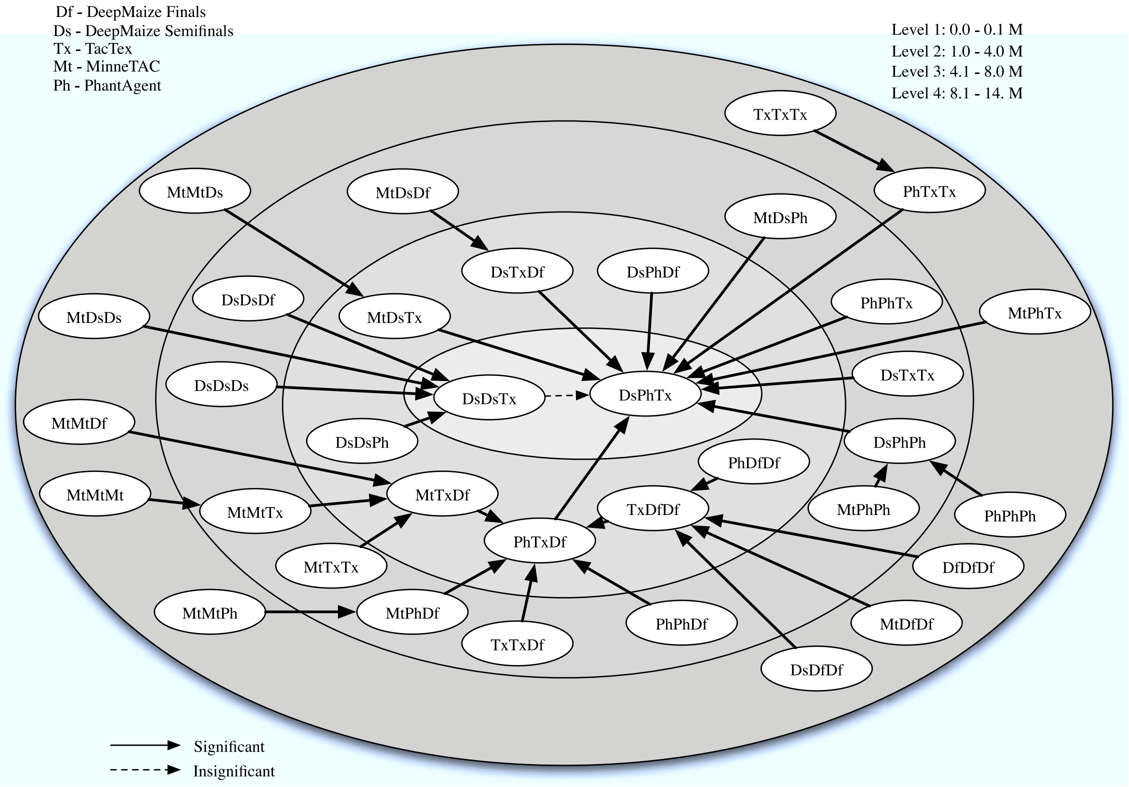


Figure 2.2 Deviation and profile stability analysis for the 2006 tournament agents.

Agent	% Positive Deviations	Best Deviation	Mean Deviation	Std. Error
TacTex	61.67	18	1.45	5.91
Deep Maize SF	63.33	5	1.43	4.68
PhantAgent	60.00	8	0.89	4.77
Deep Maize F	53.33	3	0.88	4.62
MinneTAC	11.67	0	–4.67	6.41

Table 2.2 Deviation statistics for the game containing SCM 2006 tournament agents.

DeepMaize SF are beneficial in at least 60% of the cases. The mean value for deviating is highest for TacTex and DeepMaize SF, and TacTex is the best deviation most frequently. The three agents comprising the Nash equilibrium profiles are nearly indistinguishable in this analysis.

A striking feature of the game data is that agents tend to perform poorly against exact copies of themselves. This affects all agents to some degree, though the magnitude does vary across different agents. All of the homogeneous profiles that consist of a single strategy used by all players are among the least stable pure strategy profiles (they are in the outer rings in Figure 2.2). We observe the same phenomenon in a similar analysis of agents from the 2005 tournament (Jordan et al., 2007). An intuitive and interesting explanation for this phenomenon is that it reflects a congestion effect, in which agents compete to fill the same market niches and exploit the same opportunities. Multiple copies of the same agent will tend to concentrate in the same market niches, lowering profits due to the increase in competition. Some of the forms of interference between identical agents may be due to relatively domain-specific or agent-specific issues, such as using particular lead times for ordering components or bidding on customer orders in a particular manner. Other forms of interference are more general, such as making similar predictions and using similar weightings of production across market segments.

The tendency for identical agents to perform poorly does raise some questions for our analysis, and particularly the use of player reduction as an approximation of the game. Player reduction makes our analysis especially sensitive to this issue, as agents will always play against at least one copy of the same agent. Presumably, TAC entrants design their agents with tournament play in mind, and so may not be concerned about the performance of their agents with copies of themselves in the environment. It is not surprising that agents not designed with self-play in mind perform relatively poorly in this situation. However, there are also reasons why self-play may be an important indication of agent performance that is neglected by tournament analysis. Agents that perform poorly in self-play may be particularly susceptible to imitation, and may not have a sustainable advantage over competitors as others incorporate successful elements of the strategy into their own agents.

Another interesting result of the 2006 game analysis is that the semi-finals version of Deep Maize outperforms the version from the final round. We based our decision of which version to use on intuitive strategic reasoning, making informed guesses about what the other finalists were likely to do, and further guesses about what the best response to this strategic environment would be. The evidence from our game-theoretic analysis strongly suggests that this reasoning led to the wrong choice between these alternatives. While it would not have been possible to gather the exact data presented here before the tournament since many of the agents in the analysis were not released until after the tournament, it would have been possible to conduct similar experiments using older tournament agents and other variations of Deep Maize. The empirical evidence from these experiments would have given us a firmer basis for reasoning about the alternative strategies, and may have allowed us to avoid the apparent mistake in selecting the version that appeared in the final round. This is anecdotal evidence, but it does suggest that there is potential for guiding strategy selection in the SCM game using empirical game-theoretic methods.

2.3.2 Analysis of Combined 2005 and 2006 Agents

One of the exciting opportunities afforded our empirical analysis methodology is to consider combinations of agents not observed during tournament play, including agents from different years. In this section we consider eleven tournament agents available from the repository that participated in either the 2005 or 2006 TAC SCM competition, all of which are listed in Table 2.4. The data set for this analysis consists of roughly 10,000 simulated games using both the 2005 and 2006 SCM game servers. There were two minor rule changes between 2005 and 2006. Under 2006 rules, the identity of opposing agents is revealed at the start of the game, and the reputation mechanism for suppliers was slightly modified. Agents from 2005 are compatible with the new server, but may be at a disadvantage because they were not designed for the new rule set.[8]

One of the interesting questions we can ask using this data set is whether agents are improving in subsequent competitions. A test for this hypothesis is whether the strongest 2006 agents are beneficial deviations from the equilibrium context of the game restricted

[8] Comparisons of identical profiles for which we have data for both the 2005 and 2006 servers show small but statistically significant differences in performance.

Background Context		Deviation Gain (ε)		
05 Agent	Mixture		Server Rules	
Deep Maize	0.083	06 Agent	2005	2006
Mertacor	0.431	PhantAgent	5.33M	6.57M
PhantAgent	0.314	TacTex	5.07M	4.73M
TacTex	0.172	Deep Maize SF	4.22M	4.56M

Table 2.3 Deviation benefit comparison for top 2006 agents in the context of a symmetric mixed Nash equilibrium of top 2005 agents, using both 2005 and 2006 server rules.

to include only 2005 agents. We find a symmetric Nash equilibrium for the 2005 agent set including Deep Maize-05, Mertacor-05, PhantAgent-05, and TacTex-05.[9] Using this equilibrium of 2005 agents as the background context, we test deviations to three of the top 2006 agents. The results are given in Table 2.3, along with the probabilities of each 2005 agent in the equilibrium context. We repeated this experiment using both the 2005 and 2006 game servers to run the simulations. Each of the 2006 agents is a beneficial deviation from the 2005 equilibrium, regardless of the server rules. This evidence offers strong support for the hypothesis that the top agents improved from 2005 to 2006. In the subgame that includes the background agents and Deep Maize-06 SF, Deep Maize-06 SF is the sole survivor of iterated elimination of dominated strategies, providing even stronger evidence for improvement in this agent.

We extend idea of testing deviations from a particular background context to provide an alternate means of ranking agents. Specifically, we rank agents according to the payoff they would receive as a deviation from a symmetric mixed-strategy equilibrium. This ranking supplements the tournament results in that it provides a complete ordering and facilitates comparisons between arbitrary sets of agents, including agents that participated in different years of the competition. Many of the experimental designs used by TAC participants to test agent variations share the idea of testing against a fixed background context; the key distinction we make is using a game-theoretic criteria to select *which* background context is

[9] The support of the mixed strategy equilibrium of the full set of 2005 agents we have data for also includes MinneTAC. This agent has very low probability in the equilibrium, and omitting this agent reduces the data required for our analysis considerably.

Agent	Deviation Gain	Tournament Scores	
		Finals 05	Finals 06
TacTex 06	0	n/a	5.85
PhantAgent 06	0	n/a	4.15
Deep Maize 06 SF	0	n/a	n/a
Mertacor 05	–0.57	0.55	n/a
Deep Maize 06 F	–0.95	n/a	3.58
Maxon 06 S	–1.03	n/a	n/a
MinneTAC 05	–1.23	–0.31	n/a
PhantAgent 05	–1.51	n/a	n/a
Deep Maize 05	–3.18	–0.22	n/a
MinneTAC 06	–3.48	n/a	–2.70
TacTex 05	–5.96	4.74	n/a

Table 2.4 Ranking of eleven TAC SCM agents based on deviations from an equilibrium context, coupled with tournament results (in $M).

most relevant.

In Table 2.4 we present a ranking of eleven agents from 2005 and 2006 in the context of the symmetric mixed Nash equilibrium of the 2006 game.[10] The equilibrium is a mixture of TacTex, PhantAgent, and Deep Maize SF, with probabilities [0.254, 0.188, 0.558]. All agents are ranked based on the benefit of deviating to the agent from the equilibrium context. When applicable, the table also lists the tournament results for the agent (final round only). None of the agents that do not have support in the initial equilibrium profile has a positive deviation benefit, implying that the profile remains an equilibrium in the full game containing all eleven of these agents.

The results of this ranking provide further support for the hypothesis that agent performance was significantly improved in 2006 over 2005. Previously, we provided results indicating that the 2005 equilibrium is not robust to the addition of any of the top 2006 agents; all are beneficial deviations. The 2006 equilibrium is robust to the addition of *any* of the 2005 agents, as any of these deviations typically incurs a large loss. Mertacor-05 (Kontogounis et al., 2006) is an exception, with a loss less than two of the 2006 agents

[10] This is the only symmetric equilibrium we have found after extensive search with several methods, but we cannot guarantee that it is unique.

(including Deep Maize-06 F which placed third overall). In the more detailed study of 2005 agents, Mertacor-05 demonstrated exceptionally strong performance in the game-theoretic context—significantly better than we would have predicted based on its tournament performance. This agent appears to have good performance in self-play, relative to the other agents in our pool. Overall, the rank ordering given by this method generally corresponds to the tournament rankings, where they are available. An exception to this is TacTex-05, which ranks lower than one might expect based on tournament performance.

2.3.3 Discussion

This case study illustrates many of the challenges of analyzing complex multi-agent systems. Exhaustive evaluation of TAC SCM strategies is infeasible due to the massive strategy space and high cost of computing the payoff table. Nevertheless, we are able to gather useful information about pertinent strategic question in this domain by combining empirical methods with game-theoretic analysis. The results presented in this chapter are a sample of the results we have gathered in this domain, which include:

- Evidence of a congestion effect for playing identical strategies in the TAC SCM game—a potentially interesting aspect of agent performance that is not evaluated by tournament play (Jordan et al., 2007).
- Empirical support for the claim that the strongest tournament agents improved between the 2005 and 2006 tournaments (Jordan et al., 2007).
- Additional insight into the performance of several agents, including Mertacor-05 and PhantAgent-05, both of which demonstrated stronger performance in our analysis than anticipated by the tournament results (Jordan et al., 2007).
- Verification of a rational basis for extreme early procurement behaviors observed in the first TAC SCM competition, and analysis of a preemptive strategy that improved profits across all manufacturers by preventing this behavior (Wellman et al., 2005a).
- Evidence that no (reasonable) setting of a storage cost parameter introduced in 2004 to reduce early procurement incentives would have been likely to achieve the desired effect of substantially reducing the level of early procurement in the game (Vorobeychik et al., 2006).

Taken together, these results comprise part of a growing body of evidence for the utility of empirical game-theoretic analysis in analyzing very complex games. Useful results have also been achieved in several other domains using similar methods; a review of these studies is presented in related work. With feasibility established, the next set of questions concern the best methods for conducting this analysis. To date, there has been relatively little work directly evaluating different approaches for building and analyzing empirical game models, though some studies are beginning to emerge (Walsh et al., 2003; Wellman et al., 2005b; Jordan et al., 2008; Vorobeychik and Wellman, 2008). In the following chapters, I will present results that further the agenda of developing and evaluating methods for empirical game-theoretic analysis.

2.4 Related Work

The TAC SCM game has inspired an extensive literature that includes descriptions of individual agents (many of which are referenced above), analysis of components of agent strategies, and broader analysis of the tournament and comparisons among agents. Arunachalam and Sadeh (2005) and Eriksson et al. (2006) provide broad overviews of the scenario along with descriptions of early tournaments. Kiekintveld et al. (2006c) provides more detailed analysis of the 2004 tournament, including a description of important strategic developments during that year, and Ulrich (2006) analyzes the 2005 tournament finals. There are several instances of new analysis techniques developed for this domain, including methods for variance reduction (Sodomka et al., 2007), measures of the bullwhip effect (Jordan et al., 2006), and a tool for evaluating strategies by simulating qualitatively different market conditions (Borghetti et al., 2006). Researchers have also used game theory to model specific aspects of supply chain decision making (Cachon and Netessine, 2004). Zhang et al. (2004) apply this approach to TAC SCM, creating stylized models of parts of the game. There is also a body of work that applied machine learning methods to learn market prices in the TAC SCM game (Pardoe and Stone, 2007; Ketter, 2007; Kiekintveld et al., 2008). While not explicitly game-theoretic, there is a sensitivity to strategic interactions in this work, as the prices (and thus, correct predictions) will generally depend on the opposing agents. The strategic nature of the domain is accounted for in different ways, including online adaptation within game instances and specialized training procedures that select data based on the opponents.

The general methodology of empirical game-theoretic analysis is a synthesis of a growing body of work that applies empirical methods to study a variety of complex strategic interactions. There is a common structure to these analyses, though they employ diverse techniques for specific stages in the analysis. One of the earliest examples of using simulation to generate an empirical estimate of a payoff matrix is found in work by Walsh et al. (2002), who studied strategies for both continuous double auctions and an automated dynamic pricing game. The methodology described here is based largely on the version developed by MacKie-Mason, Osepayshvili, Reeves, and Wellman in their papers on empirical game-theoretic analysis of a market-based scheduling domain (MacKie-Mason et al., 2004; Osepayshvili et al., 2005). Reeves (2005) further develops the methodology, and presents additional applications to first-price sealed bid auctions and the TAC travel shopping game. Wellman et al. (2007) present further empirical game-theoretic analysis in the TAC travel domain, with the explicit goal of selecting a version of their agent to play in the tournament. A similar empirical methodology is used by Phelps et al. (2005, 2006) to search for new strategies in a continuous double auction domain. One of the unique aspects of the approach pursued by Phelps et al. is the use of evolutionary methods to search the strategy space and motivate solution methods. In one of the few examples of empirical game theory in a non-auction domain, I applied the methodology to study a 4-player variant of chess called chaturanga (Kiekintveld et al., 2006b).

There are also several examples of empirical methods applied to mechanism design problems. One of the first examples comes from Cliff (2003), who used evolutionary methods to search spaces of continuous auction mechanisms. Vorobeychik et al. (2006) introduced an approach using game-theoretic solution criteria, and applied this to study the relationship between settings of a storage cost parameter in the TAC SCM game and agent behavior.

Vorobeychik et al. (2007a) significantly refined this approach, and applied it to mechanism design problems for both Myerson auctions and vicious auctions. This approach has also been applied to study dynamic strategies for sponsored search auctions (Vorobeychik and Reeves, 2008). Roth (2002) advocates a combination of theory and experimentation for mechanism design, and presents several case studies of this approach applied to the design of real market mechanisms.

Most of the word referenced above use empirical evidence to directly estimate payoffs, and uses these estimates to derive a game model for analysis. There is also a line of work that uses econometric methods to build empirical models of games from observations of actions (Bresnahan and Reiss, 1991; Lise, 2001; Bajari et al., 2007; Slade, 1995). The goal of this work is typically to infer the payoff structure of the game based on observations of actions taken by a set of players, based on the assumption that the observed play represents an equilibrium, providing a justification for the observed behavior.

There are also many examples in the literature where experimental evidence is used to analyze multi-agent systems, but without applying game-theoretic solution criteria in the evaluation stage. One of the most common methods of evaluating strategies in domains where exhaustive analysis is infeasible is to use tournaments to identify the best strategies, or to rank strategies based on paired comparisons. Tournaments and rankings are ubiquitous in competitive athletics (Stefani, 1997; Park and Newman, 2005), as well as more cerebral pursuits including chess (Glickman, 1995). Within the game theory community, a famous example of using a tournament to evaluate strategies was the prisoner's dilemma tournament organized and analyzed by Axelrod (1984). The TAC competitions are a prominent example of this approach within the multi-agent systems community (Wellman et al., 2003).

Chapter 3

Strategy Selection for Empirical Games: The Meta-Game Model

The integration of empirical methods and game-theoretic analysis has yielded new approaches for analyzing complex multi-agent systems. I described one application of this approach in Chapter 2, and there are several additional examples in related literature that share a similar methodology. However, little is known about how best to implement the specific stages of the analysis. Different methods have been proposed for guiding exploration of the strategy space, building game models from empirical data, and analyzing the resulting model. The question of which methods are most effective has received little direct attention in the literature, but is key to further development of the methodology. It seems unlikely that there will be a single best method for empirical game-theoretic analysis. Rather, different approaches may be desirable depending on various factors, such as characteristics of the game (e.g., structural properties), the amount of computational power available for analyzing the game, and the forms of information available. In this chapter I develop a framework for evaluating methods of empirical game-theoretic analysis across ranges of conditions, varying these and other factors. This framework may also be used to test general relationships between characteristics of the model and characteristics of the best solution methods.

One of the most obvious reasons to analyze a game is to participate in the game and choose a good strategy. I establish an evaluation framework based on this objective. Specifically, I consider the *strategy selection task*: for a given player in a game, select a strategy to play that will result in a high payoff. There are other analysis tasks that appear in the literature, often closely related to strategy selection. One common task is to identify one or more Nash equilibria of the game. This is related to selecting a good strategy, since an equilibrium specifies that all players play a best-response, given the strategies of the other players. However, identifying an equilibrium does not necessarily solve the strategy selection task. One problem is that there may be multiple equilibria, leading to a selection problem. I discuss some further difficulties that arise when players are uncertain about the game later in the chapter.

In this work my interest is in procedures for selecting good strategies using empirical game models. These models represent estimates of the game, based on limited evidence about payoffs derived from simulations or other sources. The implication of this is that players must deal with uncertainty about the game when choosing a strategy. There are at least two general sources of uncertainty: (1) payoffs may be estimated based on a noisy sampling process, and (2) resource limitations may prevent players from gathering any direct evidence at all about the payoffs for some strategy profiles. Both of these were factors in the analysis of the supply-chain game presented in Chapter 2. The problem of uncertainty is not limited to models derived from simulation data. Weibull (2004) argues that even in very idealized laboratory settings, human subjects may have legitimate uncertainty about the true nature of the game they are playing. While I focus here on the implications of uncertainty for empirical game modeling, the results are relevant to some degree to additional settings where uncertainty is present.

I begin by presenting a model in which strategy selection algorithms are applied to empirical game models. The important insight is that these algorithms can be modeled as strategies in a larger "meta-game" that endogenizes the observation process. This model motivates a framework and performance criteria for evaluating these algorithms using game-theoretic solution concepts. I conclude the chapter by discussing a simple experiment which both exemplifies the evaluation framework and demonstrates that the uncertainty captured by the model has a powerful effect on strategy selection.

3.1 The Meta-Game Model

I propose a model for empirical game-theoretic analysis applied to the strategy selection task. The premise of the model is that each player makes a strategy choice based on a separate noisy observation of an underlying game. The full game that incorporates this observation process is the *meta-game*. The sense in which this is a meta-game is that it is a more abstract game that describes the process of reasoning about another game—there is a game within a game.

This model relates to the situation agent developers face in the Trading Agent Competition. Teams often run private experiments to test potential variations of their agent before the tournament.[1] It is not possible to explore the full space of strategies in this way, and it is unlikely that teams will focus on exactly the same regions of the strategy space. Each team has their own set of data about the game, which can be an important source of evidence for deciding on a final agent strategy. The key question is how best to interpret the limited evidence available.

3.1.1 Formal Specification

The essence of the meta-game model is a standard normal-form game embedded in an empirical observation process, as depicted in Figure 3.1. I distinguish three notions of a game in the model:

- The *base game* (g) represents the true underlying game.
- The *empirical games* (ω_i) represent each player's imperfect observation of the base game.

[1] In TAC tournaments there are also some shared observations, since early rounds are used for testing and evaluation purposes. Many teams do engage in significant private testing and intentionally hide key strategies until the final rounds (Kiekintveld et al., 2006c).

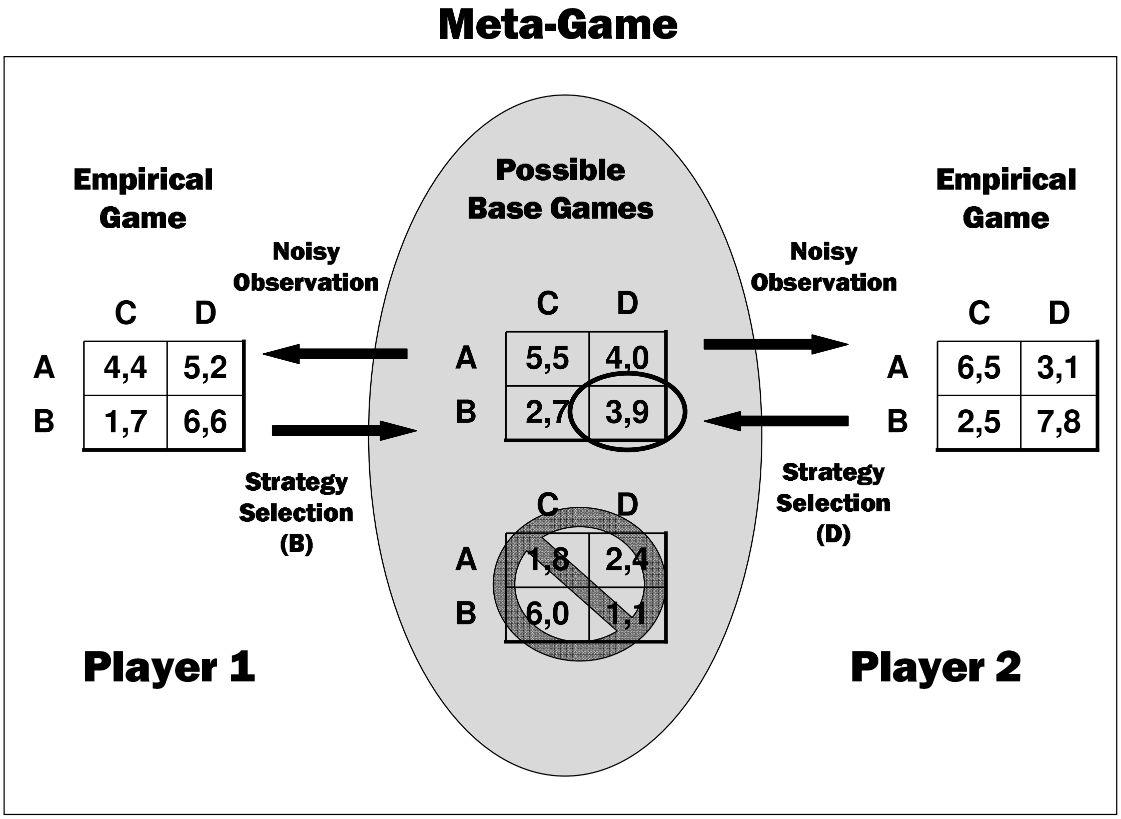


Figure 3.1 A depiction of a meta-game showing selection of the base game, generation of empirical games, and strategy selections based on the empirical games. Here, player 1 chooses strategy B and receives payoff 3, while player 2 chooses strategy D and receives payoff 9.

- The *meta-game* (Γ) encompasses the full interaction between players, including observation of the base game and strategy selection.

Formally, I define a meta-game by the tuple $\langle I, S, G, f, \{\Omega\}, \{\Theta\}\rangle$. The base game and empirical games are both standard normal-form games, with identical sets of players, I, and strategies S. There is a set of possible base game payoff functions, G, representing a class of games the players may be playing. The instances of this class, $g \in G$, correspond to the payoff functions U in a normal-form game. An instance g is drawn randomly from the class of possible games according to the density f, but is not revealed directly to the players. Instead, players receive private observations from the set of possible empirical game payoff functions, $\omega_i \in \Omega_i$. The empirical games selected for the players are related to the base game by the conditional probability function $\Theta_i = \Pr(\omega_i | g)$. In practice, I will often define the observation model ($\{\Omega\}$ and $\{\Theta\}$) implicitly by describing a procedure used to generate empirical games from base games. One natural example is adding Gaussian noise to the payoffs of a base game to generate empirical observations.

Players select mixed strategies σ_i for the base game depending on their observations. The expected payoffs they receive are determined by the profile of strategies selected, σ, and the base-game payoffs, g. The entire scenario is played once, so players cannot carry over knowledge of the game or specific opponents for future decisions.[2] It may be useful to think of a meta-game as proceeding in phases:

1. **Game Selection:** Nature draws the base game payoffs $g \in G$ according to f.
2. **Observation:** Nature generates the empirical games $\omega_i \in \Omega_i$, conditioning on g according to Θ_i.
3. **Strategy Selection:** Players select mixed strategies σ_i, conditioned on ω_i.
4. **Payoff Allocation:** Players receive expected payoffs depending on σ and the payoffs for g.

Strategies for playing the meta-game are called *meta-strategies*, and are essentially strategies for selecting base-game strategies (hence the term "meta"). Specifically, a meta-strategy $\psi_i \in \Psi_i$, maps any empirical game observation into a base game strategy, $\Psi_i : \Omega_i \rightarrow \Sigma_i$. An important insight of the meta-game model is the intuition that solution algorithms for the strategy selection task can be modeled as meta-strategies. Any algorithm that takes as input an empirical game description and returns a strategy is a valid meta-strategy. For example, an algorithm that finds a Nash equilibrium and plays according to this strategy is a meta-strategy.[3] Another valid meta-strategy selects the uniform mixed strategy regardless of the observation. Of course, these two meta-strategies may have very different performance characteristics.

I denote the payoffs for meta-strategy ψ_i in the profile ψ using the function $\Pi_i(\psi)$. The notation $\psi_i(\omega_i)$ refers to the strategy chosen by the meta-strategy for the given observa-

[2] I use a one-shot setting in part to make a clean distinction between observing the game parameters and observing a specific opponent's strategy. In a repeated setting it may be possible to learn about both simultaneously, raising many additional issues beyond the scope of this work.

[3] In the case where there are multiple equilibria, a selection method is necessary to distinguish among them.

tion. The function ψ is defined by the expected payoffs for the strategies chosen by the meta-strategies, where the expectation is taken over the distribution of base games and observations:

$$\Pi_i(\psi) = \sum_{g\in G}\sum_{\omega\in\Omega} \Pr(g)\cdot\Pr(\omega|g)\cdot u_i^g(\psi_1(\omega_1),\ldots,\psi_n(\omega_n)). \tag{3.1}$$

Here, u_i^g is the payoff function for player i in game g. Using this definition of meta-strategy payoffs, I define meta-strategy equilibrium analogous to Nash equilibrium:

Definition 7 *A* meta-strategy equilibrium *is a meta-strategy profile ψ in which no player has an incentive to deviate to a different meta-strategy. That is, $\Pi_i(\psi_i,\psi_{-i}) \geq \Pi_i(\psi_i',\psi_{-i})$ for all players i and all meta-strategies $\psi_i' \in \Psi_i$.*

The essence of this definition is a Nash equilibrium of a normal-form representation of the meta-game. A meta-strategy equilibrium is stable in the same sense that a Nash equilibrium is; given the meta-strategies used by the other players, no player would wish to change to a different meta-strategy. For this property to hold globally, it must hold for each individual observation. That is, for every possible observation, the equilibrium meta-strategies must select a best-response strategy in expectation, given the distributions of games, opponent observations, and opponent choices for each observation. If there were a better response to an observation, then the meta-strategy that modifies only the response to this observation but holds the rest of the choices fixed has a higher expected payoff, violating the equilibrium assumption.

The definition of the meta-game model presented thus far abstracts out common elements of empirical game-theoretic analysis, and makes clear the mapping between strategy selection algorithms and meta-strategies. In the sequel, I explore various instantiations of this model, defining particular classes of games and observation models. All of these variants share the high-level context of empirical observation described in this section. One further item which I have not addressed in the model is what information players have about the underlying game and observation models (e.g., whether they have common knowledge of the class of games or the way empirical games are generated). This issue bears further discussion, which I defer to Chapter 4. In principle, players may have beliefs about the underlying model. However, it is often of interest to consider restricted meta-strategies that do not make use of these beliefs.

3.1.2 Discussion

Meta-games model situations in which players do not have common knowledge of the payoffs in the game. They can be viewed as a form of Bayesian game in which the types are defined by the possible empirical game observations. Under this interpretation, meta-strategy equilibria are a form of Bayes-Nash equilibria. The type space of a meta-game is very complex, relative to most Bayesian game models found in the literature. The empirical games are multi-dimensional, and the sheer number of potential observations in realistic settings is immense. Empirical games also have a mutual dependence on the base game, so the observations are not independent. That is, each player's type reveals information about the realization of the other players' types. This must be accounted for in belief updates, making these updates considerably more difficult. In the literature, games modeled using the Bayesian framework typically have independent type distributions, though a notable exception is common value auctions (Milgrom and Weber, 1982; Chatterjee and Harrison, 1988).

One interesting limiting case of the meta-game model occurs when players directly observe the base game with no uncertainty. In this case, we might expect meta-strategies that play according to Nash equilibria to form equilibria of the meta-game. This is true *only* if the meta-strategies always coordinate on the same Nash equilibria of the base game. Consider a meta-strategy that finds all pure-strategy equilibria of the empirical game and selects a random equilibrium to play. In the game shown in Table 3.1, this meta-strategy would receive a very low payoff playing against itself. There are two pure-strategy equilibria of the

	L	R
U	–M, –M	2 2
D	1, 1	–M, –M

Table 3.1 An example game in the normal form, where M represents a large number.

game, [U,R] and [D,L]. Randomly selecting between these induces a mixed strategy. If both players use this mixed strategy, they receive arbitrarily low expected payoffs (depending on the value of M) because the mixtures place positive probability on the profiles [D,R] and [U,L]. Moreover, the profile where all players use this meta-strategy is clearly *not* an equilibrium of the meta-game. Always playing either U or R gives a higher payoff than a random mixed strategy when the other player is playing a random mixture, so this is a beneficial deviation.

This example illustrates why simply playing according to some equilibrium is not sufficient to generate equilibrium meta-strategies. Meta-strategies must also coordinate on a particular equilibrium. For instance, a meta-strategy that always selects and plays the [U,R] equilibrium *does* form an equilibrium of the meta-game when all players use it. There are many possible coordination mechanisms that would suffice for this purpose, such as selecting the equilibrium with the highest aggregate payoff.

Observation noise makes it more difficult for meta-strategies to coordinate on equilibria of the base game for two reasons. First, noise affects the perceived benefit for each of the possible deviations in the game. Profiles that are not equilibria of the base game may appear to be equilibria in the empirical games, and vice versa. This effect can be bounded if the noise is bounded by a small constant c, such that the payoffs observed in the empirical game differ from the base game payoffs by no more than c. Consider a pure profile in an empirical game with regret $\hat{\varepsilon}$. This profile's regret is at most $\hat{\varepsilon} + 2c$ in the base game, since there are two payoffs that define the maximal benefit to deviating. If $\hat{\varepsilon} \leq -2c$, the profile must be a Nash equilibrium of the base game. Furthermore, the other players' observed regret for this profile must lie in the range $[\hat{\varepsilon} - 4c, \hat{\varepsilon} + 4c]$.

A more subtle effect is that noise can break coordination mechanisms that are based on payoffs. Looking again at the game in Table 3.1, suppose that empirical games are generated by adding noise U[–5,5] to each payoff, and that $M \gg 5$. Both [L,D] and [U,R] are equilibria in any empirical games generated using this observation model. However, the coordination mechanism of selecting the equilibrium with the greatest aggregate payoff would fail here, since either profile could have the highest observed aggregate payoff. When the meta-strategies fail to coordinate on an equilibrium, the outcome can be arbitrarily bad. By breaking coordination, even small amounts of observation noise can have a very large impact on the payoffs achieved by a meta-strategy.

3.1.3 Meta-Games with Directed Observations

So far, I have assumed that players receive game observations in the form of estimated payoff matrices. This clearly abstracts away from many of the details of gathering evidence and estimating payoffs from more primitive data. Players may have some degree of control over the payoff information they observe, enabling them to use directed exploration methods. It may also be possible to make better payoff estimates by processing the raw data, for instance, by applying variance reduction methods (Wellman et al., 2005a) or machine learning (Vorobeychik et al., 2007b). Capturing these details requires a richer observation model than presented above. I describe (informally) an alternative model that allows players to direct the simulation process for estimating the game. More detailed treatments of similar simulation-based models are given by Vorobeychik et al. (2006) and Vorobeychik and Wellman (2008).

Suppose that instead of observing an estimated payoff matrix, players have access to a black-box simulator. This simulator has the capability to return a sample of the payoffs for any strategy profile in response to a query. The payoff samples may be either deterministic or stochastic, depending on the application. Players are allowed to execute only a limited number of queries before making their final strategy choice.[4] These are the same basic capabilities provided by the TAC SCM simulator used to generate game data in Chapter 2.

Players can build estimated payoff matrices using this simulation model. The key difference is that they can *choose* which profiles to sample, and may be able select profiles that are more likely to factor into the final decision. Each player faces a sequence of choices, each specifying a profile to sample. At each iteration, the choice may depend on the history of observations, so the number of possible ways to choose samples grows exponentially in the length of the sequence (i.e., the number of samples allowed). These choices are also (in principle) strategic, in that they may depend on beliefs about the other players—beliefs which must be updated after each sample to reflect the new information. Conceptually, the meta-game model could be extended to include this sampling process.[5] I do not present a formal extension here as it would add little to the discussion by itself.

Identifying an optimal policy for making these sampling choices is quite daunting, given the complications described above. However, it may be possible to identify heuristic approaches or approximations which perform well in practice. I take this approach in Chapter 5, studying heuristic policies for choosing which profiles to sample next, given access to a game simulator.

3.2 Empirical Evaluation of Meta-Strategies

My primary interest in this work is to develop practical methods that advance the state of the art for empirical game-theoretic analysis. The usefulness of any particular method depends in part on how well the method generalizes to different applications. In terms of the meta-game model, characteristics of the game class and observation model are likely to vary depending on the application. For instance, in some cases the game class may have

[4] An alternative approach to a fixed bound is to model the cost of each sample explicitly.

[5] A common way to model sequential choices in games is the extensive form (Fudenberg and Tirole, 1991); a formal extension of meta-games to include the sampling choices could be based on this formalism.

structural properties that can be exploited in the analysis. Another important variable is the amount information available about the game, which may depend on the degree of sampling noise, the cost of running simulations, or other factors. There is not a single meta-game that is a good model for all applications of empirical game-theoretic analysis—the details depend on the domain in question. Solving any specific meta-game is of limited value unless the solution generalizes in some way. The focus of my evaluation methods is to identify solutions that perform well across a range of conditions, even if they are not optimal solutions to any specific case.

Broadly speaking, there are two approaches for analyzing meta-games. The first is to find exact solutions to models that are amenable to mathematical analysis. This approach potentially offers the sharpest and most satisfying results. However, the models which can be solved may be limited in scope. Meta-games present significant challenges for exact analysis, a point I discuss further below. The general question raised by solving restricted cases is whether the results will generalize to more realistic conditions. In other words, do the solutions perform well in practice, and which are best under different circumstances?

The other high-level approach is to exploit computational tools to study more ambitious models empirically. While there are limits to the claims that can be made using this type of evidence, this approach offers a way to address questions that are not easily amenable to mathematical approaches. One use of empirical evidence is to identify the best known meta-strategies for specific meta-games of interest where exact solutions are not known. Empirical methods are also well-suited to studying variations of the meta-game model, which is useful for studying the kind of generalization described above. By comparing candidate methods across different conditions, it is possible to assess their relative strengths and weaknesses. I provide a concrete experiment demonstrating this form of evaluation in Section 3.3, showing how the performance of a naïve equilibrium prediction method is affected by varying levels of observation noise.

These broad approaches are complementary, and yield different sorts of insights. One way that they can be used in conjunction is to develop candidate meta-strategies using exact analysis of simplified models, and to evaluate these candidates over a broader range of conditions using empirical methods. There already exist a number of methods in the literature which can be adapted into candidate meta-strategies, some which have already been applied to analyze empirical game models. What is lacking is a common framework for comparing these methods. I develop an empirical framework for evaluating candidate meta-strategies in Section 3.2.2, after some additional discussion of the difficulties of solving meta-games exactly.

3.2.1 Challenges for Finding Exact Solutions to Meta-Games

A natural goal is to identify exact equilibrium solutions for specific meta-games. As described above, meta-games are games of incomplete information, and they are particularly challenging instances in several respects. The most problematic is that players receive very complex information in the form of empirical game estimates. Solving incomplete information games is very hard in general. Computational complexity results offer one indication of this. Constructing an equilibrium for complete information games is PPAD-complete (Daskalakis et al., 2006). Incomplete information games contain complete information games as a special case,[6] so constructing an equilibrium is PPAD-hard. Determining whether a pure-strategy Bayes-Nash equilibrium exists is NP-complete, even for 2-player, symmetric incomplete information games (Conitzer and Sandholm, 2003). For arbitrary incomplete information games with infinite type spaces, closed-form solutions may not exist. There are no general-purpose algorithms available for computing exact Bayes-Nash equilibria in this case, though methods of approximating equilibria have recently been developed for some classes of games (Armantier et al., 2007; Reeves and Wellman, 2004).

Another point of reference is the literature on *global games* (Carlsson and van Damme,

[6] A complete information game corresponds to an incomplete information game with a single type for each player.

1993), which are closely related to meta-games. Global games are also games of incomplete information, and embed an underlying game within a noisy observation process. Players receive observations that are determined by adding random noise terms to each payoff, and do not have common knowledge of the payoffs. The key theoretical results show that there is a unique equilibrium of the global game as the noise terms vanish in the limit, effectively selecting one of the equilibria of the complete information game (Carlsson and van Damme, 1993; Frankel et al., 2003). There are applications of this model to bank runs, debt pricing, and similar scenarios (Morris and Shin, 2003). However, there are no general theoretical result characterizing the equilibria of global games away from the limit. Carlsson and van Damme (1993) note in their own discussion of the model that it is mathematically challenging beyond the class of 2-player, 2-action games.

3.2.2 Experimental Framework

Experimental analysis offers a way to supplement theoretical analysis of meta-games, the scope of which is limited by the complexities of the model. I propose a framework for benchmarking candidate meta-strategies that adapts many aspects of empirical game-theoretic analysis, as outlined in Section 1.3. To apply the framework, the meta-game model must first be instantiated by specifying the following:

- The class of possible base games
- The distribution from which base games are drawn
- The method used to generate observations for players

In addition, I specify a restricted set of candidate meta-strategies for analysis. These candidates may be algorithms drawn from the literature, solutions of simplified models, or heuristics. Given a set of candidate strategies, I use Monte Carlo simulation to estimate the payoffs of the constrained meta-game. To do so, all of the components must be implemented in computer code; this requires that the game class and observation model are generative. Meta-strategies must be valid (return a strategy) for all possible observations. Simulation to estimate payoffs follows the stages outlined in Section 3.1.1. These stages are repeated for all possible combinations of meta-strategies to estimate the full payoff matrix for the constrained meta-game, sampling many times for each strategy profile.

This payoff matrix must be symmetric if players cannot condition their choice of meta-strategy on any private information. In the meta-game, I assume that all payoff information is revealed through the observation process, so players have the same information before the observations are revealed. I use this to reduce variance in the estimated meta-game by averaging all payoffs that must be equivalent due to symmetry. The estimated payoff matrix for the meta-game is the basis for evaluating the strategy choices made by the meta-strategies. Standard game-theoretic solution concepts can be used to analyze these meta-games, including dominance, regret, and equilibrium. I introduce and motivate specific performance measures as necessary.

What types of questions is this experimental framework suited to answering? One possible use is to select the best meta-strategy from a set of candidates for a specific meta-game. Finding a solution to a constrained meta-game does not offer any guarantees on solution quality. However, as argued in Section 2.2.1, the solution is more likely to be a global solution than an arbitrary profile, even given conservative assumptions about the quality of meta-strategies that are not includes in the set of candidates. A more interesting use of the model is to explore relationships between meta-strategy performance and characteristics of the meta-game. This has the potential to offer more general insights into how to design meta-strategies, and to identify strengths and weakness of different techniques. To facilitate this type of analysis, is it often useful to define parameters that vary gross characteristics of the meta-game or meta-strategies. For example, I later define parameters of the observation model that vary the amount of information contained in each players' observation of the game. By simulating meta-games with different settings of these parameters, it is possible to observe how the performance of meta-strategies changes in response to the amount of information available. Another example of a similar experiment is to test whether some meta-strategies are able to exploit specific structural properties of the underlying game.

There are some limitations of the empirical approach that deserve mention. First, the simulated meta-games are estimates of the payoff matrix, and may contain sampling noise. In my experiments, I mitigate this by taking very large numbers of samples, but it cannot be avoided entirely. There may also be a significant computational cost to estimating the meta-game, especially if base games are large and the meta-strategies are computationally intensive. In practice, this may limit the analysis to modest sets of candidate meta-strategies.[7] The restriction on the space of meta-strategies considered is perhaps the most severe limitation. Given this restriction, it is generally not possible to make positive global claims about the properties of the meta-strategies in the full meta-game (e.g., whether they form an equilibrium of the meta-game), though it is possible to show definitive negative results (e.g., that a profile cannot be an equilibrium). The usefulness of the analysis depends in part on careful selection of promising candidates, and an iterative process of improving the best known solution. To the extent that a candidate solution has survived many challenges, there is a stronger claim that it is likely to represent a global solution, or at least a close approximation.

3.3 Naïve Equilibrium Analysis

I begin with an experiment designed to demonstrate the effects of observation noise on naïve equilibrium analysis in a general setting. Section 3.1.2 gave a motivating example showing that noise can break the coordination necessary for equilibrium behavior, potentially leading to poor payoffs. Here I show that the effect is not limited to isolated cases, but is a pervasive problem that leads to poor performance for naïve equilibrium methods on average. The more general point is that payoff uncertainty has a dramatic effect on strategic analysis—an effect captured by the meta-game model. This experiment also serves as a relatively simple

[7] I have experimented with up to 20 meta-strategies; at this point, computational costs become burdensome using the present implementation.

example of the general experimental framework described previously, and shows the sorts of insights that can be achieved using this paradigm.

I set up the experiment using two candidate meta-strategies. The first represents a naïve equilibrium analysis that ignores uncertainty and applies the standard perfect-information solution concept of Nash equilibrium to the empirical game. *Naïve Pure Strategy Nash Equilibrium* (NPSNE) finds the pure-strategy profile of the empirical game with minimum regret ε, and plays according to this profile. The criterion of selecting the profile with minimum regret serves as a coordination mechanism, since it will identify a unique pure-strategy profile unless there are ties in the regret function.[8] In an idealized setting where pure-strategy equilibria exist and there is no observation noise, the NPSNE meta-strategy will always coordinate on the same equilibrium of the base game. This implies that the profile where all players use this meta-strategy is an equilibrium of the meta-game in this case, since there is no possible best-response to the equilibrium strategies in the base game that could be selected by another meta-strategy. The second candidate meta-strategy, *Best-Response to Uniform* (BRU), does not attempt to make *any* use of information about the other players' payoffs to predict their behavior. It always predicts that opponents will play a uniform random mixture over their pure strategy choices, and plays a best-response to this random distribution of opponent play.

The base games are drawn uniformly from a class of 2-player, 4-strategy games with random payoffs.[9] Each payoff is drawn uniformly between 0 and 1. For the purposes of this experiment, I screen the games within this class so that all instance have pure-strategy Nash equilibria.[10] I generated 2500 sample random games, of which 1888 have PSNE. The observation model used for this experiment generates empirical games by adding independent mean-zero Gaussian noise to each payoff in the base game.

[8] This does not occur in the set of games tested.

[9] These games are generated using the GAMUT tool, and are instances of the "Random Game" class (Nudelman et al., 2004).

[10] This ensures that NPSNE will always find a true equilibrium. I observe the same qualitative result if games without PSNE are included in the experiment.

	BRU	NPSNE
BRU	0.66	0.67, 0.67
NPSNE		0.76

Table 3.2 Meta-game for NPSNE and BRU with noise standard deviation of 0.05. Only the distinct payoffs are shown in the matrix.

I test a range of different noise levels, corresponding to Gaussian noise with different standard deviations. Each noise level is formally a separate meta-game with a different observation model. The meta-games for this setup can be represented as symmetric 2x2 payoff matrices, with 4 distinct payoffs. An example meta-game payoff matrix is shown in Table 3.2. All of the payoffs in the meta-games are estimated by simulation, using 5 sample observations for each of the 1888 sample game instances.

Figure 3.2 plots the payoffs for a set of meta-games with varying levels of observation noise. Each line corresponds to one of the four payoffs in the meta-game, defined by the four possible pairings of the two meta-strategies. As argued above, NPSNE must be an equilibrium of the meta-game in the limiting case where there is no noise (on the far left of the plot). Indeed, NPSNE achieves a very high payoff when both players use it in this case, though most of the benefit is lost if the other player chooses according to BRU. The equilibrium analysis remains beneficial as small amounts of observation noise are introduced, but the performance of NPSNE relative to BRU steadily decreases as the noise level increases. Beyond the noise level of roughly 0.2, BRU is a *dominant* meta-strategy; regardless of whether the opponent is using NPSNE or BRU, the player achieves a higher payoff using BRU. This is a rather striking result, and clearly shows that ignoring payoff uncertainty in equilibrium analysis can lead to poor strategy choices. When there is sufficient uncertainty, even the extreme method of predicting completely random opponent play results in better choices than making strategic predictions based on faulty assumptions.

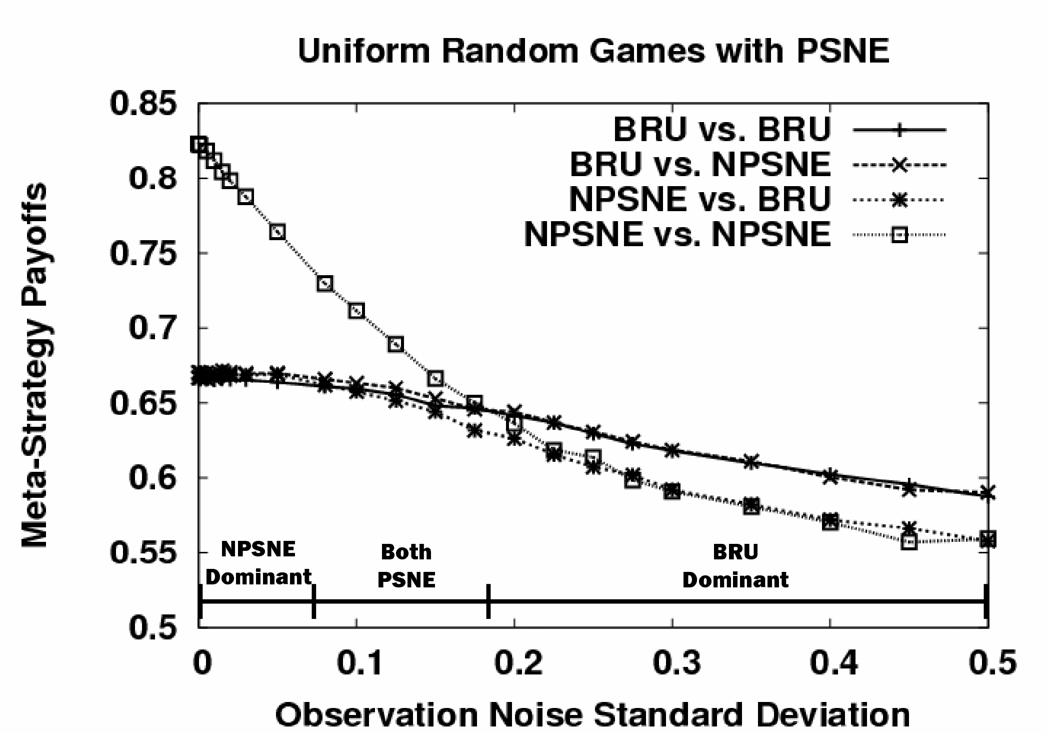


Figure 3.2 Expected payoffs for NPSNE and BRU with varying noise levels, for games with uniform random payoffs and a Gaussian observation model.

3.4 Related Work

One feature of the meta-game model is that players could be playing any game from a class of games. Bednar and Page (2007) also study a model in which players play collections of game instances (called ensembles). They consider players that are bounded-rational, and only select from a limited set of strategies represented by finite automata. Given this restriction, they may not be able to play optimally in each individual game in the ensemble. The experimental results show that players often learn to play optimal strategies for the collection of games as a whole when using evolutionary adaptation, even if they are suboptimal for particular games.

Another line of work that considers general algorithms for playing classes of games was initiated by Pell (1993). The idea is to develop algorithms which analyze a game description and generate a strategy based on the description. This is similar in spirit to my notion of a meta-strategy, and Pell actually uses the term meta-game to describe his approach. The specific algorithms developed in this work operate on a class of games similar to chess, all described in a logical language. The evaluation of this algorithm takes place in a tournament setting.

The General Game Playing competition (Genesereth et al., 2005) was inspired largely by this work, and has the goal of encouraging development of more general approaches to game analysis. The competition has been played annually since 2005. Entrants in the competition design algorithms that take input in the form of a game description given in a specified language. The games are typically variations of common board games, such as chess or othello. After some time to analyze the game description, the algorithms play the game in a tournament setting. Some examples of techniques that have been used in this competition include co-evolution of strategies (Reisinger et al., 2007), feature extraction (Kaiser, 2007), and transfer learning (Banerjee and Stone, 2007).

One of the insights of the meta-game model is that the relevant notion of equilibrium when dealing with an estimated game is an equilibrium of the meta-game, rather than an equilibrium of the estimated game. This is similar in spirit to the idea of a learning equilibrium for repeated games (Brafman and Tennenholtz, 2004). The learning equilibrium is an equilibrium between learning algorithms in a repeated game context, rather than an equilibrium of the stage game. A similar point is argued by Shoham et al. (2007), who propose the notion of targeted optimality against specific classes of opponents as an alternative to convergence to equilibrium in the stage game as a goal for multi-agent learning.

There are also many example in the literature on repeated games of using experimental evidence to evaluate potential strategies. One of the first examples is Axelrod's Prisoner's Dilemma tournament (Axelrod, 1984). Several other studies have compared the performance of small sets of candidate multi-agent learning algorithms in repeated games (Nudelman et al., 2004; Lipson, 2005; Powers and Shoham, 2004). A final interesting example is the Turing tournament proposed by Arifovic et al. (2006), designed to test how well algorithms predict the play of humans. One set of algorithms is tasked with mimicking the play of humans, while another tries to distinguish the play of humans from the algorithms. All of these examples rely on empirical evidence to evaluate candidate learning algorithms, many using the framework of a tournament. Tournaments evaluate only a limited set of the possible combinations of strategies, while the game-theoretic analysis I employ is based on analysis of all strategy profiles for a set of candidates.

Chapter 4

Selecting Strategies with Noisy Game Observations

In complex multi-agent systems, uncertainty about the game model is often inevitable. Using simulation to build empirical game models is one way to apply game-theoretic tools to very complex domains with large strategy spaces. However, empirical data may provide only payoff estimates, and even these may be unavailable for some strategy profiles. The meta-game introduced in the previous chapter models scenarios in which players have limited information about the game they are playing. In this chapter I address the question of how players should use empirical game models to guide their strategy choices. I apply the meta-game framework to evaluate three parameterized families of meta-strategies.

The meta-strategies I consider are generic in the sense that they select a strategy based on any complete estimate of a payoff matrix. All three families are motivated by equilibrium concepts for complete-information games, but generalize these concepts to approximate the effects of payoff uncertainty. Each family is defined by a single parameter that controls the amount of uncertainty expressed in predictions about how opponents will play. I show experimentally that there is a systematic relationship between observation noise and the best parameter settings. Further experiments compare the three families of meta-strategies, and identify the current champion for the strategy selection task under uncertainty. The champion is robust, in that meta-strategies from this family outperform the other families for a wide range of conditions.

4.1 Candidate Meta-Strategies

I introduce three families of meta-strategies (i.e., algorithms that analyze empirical game matrices to select strategies). They share some high-level features. All three select a best-response strategy to a prediction of how the other players will play, expressed as a probability distribution over the pure-strategy profiles. The two meta-strategies introduced in Section 3.3 (NPSNE and BRU) both have this form, but base their predictions on very different assumptions. At one extreme, NPSNE uses idealized strategic reasoning to predict other players' strategies precisely, based on payoff information and rationality assumptions. At the other, BRU predicts random opponent play, disregarding information about the other players' payoffs entirely. The experimental results presented for these meta-strategies show that the relative quality of these predictions depends on the gross level of observation noise. For low noise conditions, NPSNE dominated BRU, and for high noise conditions, the situation is reversed.

Between these extremes lies a spectrum of *distributional* predictions that factor noise into strategic reasoning in some intermediate way. The three meta-strategy families I introduce all have a single parameter that interpolates predictions between these two extremes. At one end of the parameter space, the meta-strategy uses an uninformed prediction. For parameter settings at the opposite extreme, the meta-strategies make predictions based on an equilibrium concept that assumes complete information. This spectrum of predictions is similar in spirit to the levels of reasoning in the cognitive hierarchies model introduced by Camerer et al. (2004). In this model, level 0 players play according to a uniform random distribution. Players using higher levels of reasoning best respond to a distribution of opponent play that assumes all other players use lower levels of reasoning. For instance, a level 2 player assumes a mixture of level 0 and level 1 players.[1] As the number of reasoning levels increases to ∞ play may converge to a Nash equilibrium, but this is not necessarily the case (unlike the meta-strategies considered in this work).

[1] The particular distribution is a parameter of the model. A Poisson distribution is typically used in practice.

The parameter spaces of these meta-strategies reflect an inherent tension in the prediction task. Making specific predictions (e.g., a point equilibrium) allows the player to select a best response targeted specifically to the opponents' strategies. The tradeoff is that inaccurate predictions may result in poor choices against the actual strategies chosen by the opponents. Noisy payoff observations increase the risk of inaccurate predictions, because players share less knowledge of the game. After introducing the meta-strategies, I explore this tradeoff experimentally, analyzing the relationship between observation noise and the most stable parameter settings within each family. The hypothesis is that increasing the observation noise will favor meta-strategies that predict broader distributions of play. Vorobeychik and Wellman (2006) advocated a similar idea of using distributional predictions in the context of mechanism design.

Another important aspect of the meta-strategies I study is that they do not make use of explicit beliefs about meta-game they are playing, such as knowledge of the game class. In principle, if players have common knowledge of the full description of the meta-game, they could derive explicit Bayesian beliefs from their observations and use these to implement equilibrium meta-strategies. While I do not rule out the possibility of players having such knowledge, there are several reasons that I focus on meta-strategies that do not use these beliefs. The observation process in the meta-game is intended to model the information players have about the game, and how it is revealed. Using external beliefs to make choices violates this spirit to some extent. Moreover, in practice it seems quite difficult to derive accurate beliefs about the factors that define the meta-game. For example, reading the game specification for the TAC SCM game (Section 2.1) does not give much intuition about the distribution of payoffs in the game; it is far easier to derive such information from simulation data. There are also practical advantages to meta-strategies that do not use explicit beliefs about the meta-game. They are generic, in the sense that they are valid regardless of the specifics of the underlying model. This means that they can be executed without modification for different meta-games, though the quality of the strategy selections depends on the underlying meta-game.

4.1.1 Pure-Strategy Approximate Nash Equilibrium

When players have only a noisy observation of the game, it is more difficult for them to coordinate on an equilibrium of the base game. Profiles that are equilibria of the empirical game may not be equilibria of the base game, and equilibria of the base game may not be equilibria of the empirical game. Neither an equilibrium of the base game nor of the empirical game necessarily correspond to equilibria of the meta-game.

One possible way to account for this when analyzing an empirical game is to consider a broader set of approximate equilibria. In Section 1.2.2 I define an approximate Nash equilibrium based on the notion of regret. If no player can improve their payoff by more than ε by deviating to another strategy from a profile σ, the profile has regret ε and is an ε-Nash approximate equilibrium. This approximation relaxes the requirement that all players choose an exact best response in a standard Nash equilibrium, and provides a measure of how close any given profile is to being a Nash equilibrium.

A common prediction in the game theory literature is that Nash equilibrium profiles are more likely outcomes than profiles that are not equilibria, due to their stability properties. A natural extension of this prediction is that approximate equilibrium profiles with low regret are more likely outcomes than profiles with large regret measures. This is especially compelling in the context of noisy observations, where the observed regret for a profile may not be the true regret. Profiles that have lower regret are more likely to *appear* to be equilibria to both players, and are perhaps more likely to be played as a result. Based on this intuition, I define a family of meta-strategies that uses the *ε-Nash Solver* (ENS) to predict an outcome distribution over the strategy profiles in a game. The algorithm places greater weight on approximate equilibrium profiles with low regret, explicitly including negative regret values. A Boltzmann distribution is used to construct a distribution from the regret measures:

$$\Pr(s) = \frac{e^{-\varepsilon(s)/\tau}}{\sum_{s' \in S} e^{-\varepsilon(s')/\tau}}. \tag{4.1}$$

There is one parameter of this distribution, τ, commonly called the temperature parameter. This parameter divides the ε measure for each profile, either magnifying the differences (for small values less than 1) or reducing the differences (for large values). In one limiting case as $\tau \to \infty$, the probability distribution approaches the uniform distribution, ignoring differences in regret. At the other extreme as $\tau \to 0$, all of the weight in the probability distribution is placed on the profile with minimum regret. Intermediate values of τ interpolate between these two extremes. The family of ENS meta-strategies is defined by the possible settings of this parameter. Strategy selections are made by computing a best response to the predicted distribution over strategy profiles. In the limiting case where $\tau \to 0$, the ENS meta-strategy is equivalent to the NPSNE meta-strategy from Section 3.3.

I note one subtle reason why the predictions produced by this method may contain errors. Since the probabilities are determined based on properties of strategy profiles (i.e., regret), it is possible that the predicted distribution could not literally be observed for any mixed strategy profile σ. Consider a distribution that predicts profile [1,1] will occur half of the time, and [2,2] the other half. For this prediction to be correct, players must select randomly between playing strategy 1 and playing strategy 2 using a common randomization device. Strategies that rely on common randomization are called correlated strategies (Aumann, 1987). My model does not allow correlated strategies, so predictions that require randomization are clearly making prediction errors in these cases. However, the predictions and strategy selection are still well-defined, and may be reasonable approximations of the true distribution of play.

4.1.2 Mixed-Strategy Approximate Nash Equilibrium

The previous method is based on a relaxation of the equilibrium model, but considers only pure-strategy profiles. I now define a family of meta-strategies that also finds approximate mixed-strategy Nash equilibria, using the *Replicator Dynamics Solver* (RDS). The algorithm first approximates mixed-strategy equilibria using an implementation of replicator dynamics (Taylor and Jonker, 1978) to search the space of mixed-strategy profiles. My implementation of replicator dynamics is based on the version described by Reeves (2005). While this search method is not guaranteed to find an equilibrium, in practice it almost always identifies a profile that is a close approximation to an equilibrium (i.e., has low ε).[2] The search process models an evolving population such that better strategies have greater representation in the subsequent population. Let $p_g(s_i)$ denote the fraction of the population for player i playing pure strategy s_i in generation g, $EP(s_i)$ denote the expected payoff to s_i in generation k, and W the lowest possible payoff.[3] The population update rule for each generation is:

$$p_k(s_i) \propto p_{g-1}(s_i) \cdot (EP(s_i) - W). \tag{4.2}$$

I use random restarts to prevent the search from becoming stuck in local optima. Both the maximum number of generations and how often the search restarts are parameters of the search. I fixed these for all experiments after preliminary exploration to determine settings that quickly found good equilibrium approximations.[4] The profile identified during the search with the minimum regret is the equilibrium estimate.

The ENS method uses the fact that we can quickly compute the regret for all pure-strategy profiles to make predictions about the relative likelihood of these profiles. The RDS method of searching the mixed-strategy profile space is able to identify approximate equi-

[2] In my experiments, the regret for the best profile identified using this search averaged roughly 0.3% of the maximum payoff.

[3] I do not assume symmetry in the game.

[4] In all experiments, the search ran for 1000 iterations, with forced random restarts after every 100th iteration. In addition, after every 5 iterations the search was restarted if the current profile had a regret higher than 8% of the maximum payoff.

libria, but does not provide an analogous measure to compare the relative likelihood of all mixed-strategy profiles. However, it is still possible to interpolate between the approximate equilibrium prediction and the uninformed prediction using a generic parameterization. The algorithm forms a predicted distribution over pure-strategy profiles by taking a weighted combination of two distributions: the distribution induced by the predicted equilibrium, and the uniform distribution. A weight of δ is placed on the uniform distribution, and $1-\delta$ on the equilibrium distribution. As before, the final strategy selection is made by computing a best response to this prediction. The family of RDS meta-strategies is defined by the possible settings of the parameter δ.

4.1.3 Logit Equilibrium

The final family of meta-strategies is based on logit equilibrium, which is a specific form of the Quantal-Response Equilibrium (QRE) first introduced by McKelvey and Palfrey (1995). The key idea of this equilibrium model is to imagine that players make their choices using *noisy* best-response functions. To define a noisy best-response function for player i, first fix the strategies for all other players to σ_{-i}. The conditional strategy profile σ_{-i} induces an expected payoff $u_i(s_i, \sigma_{-i})$ for each of i's pure strategies. A best-response function selects a strategy that maximizes this expected payoff. A noisy best-response function selects the strategy that maximizes $u_i(s_i, \sigma_{-i}) + \eta_{s_i}$, where η_{s_i} is a noise term added to the expected payoff for strategy s_i. The noise terms are drawn from some probability distribution, which in turn induces a probability distribution over pure strategies specifying how likely each is to be selected. Strategies that are better responses are selected more frequently, but no strategy is selected with certainty. Even a strategy with very low expected payoffs may still be chosen if the realization of the noise term for the strategy is high enough.

A *quantal-response equilibrium* is a fixed point in the mapping of noisy best-response functions. In the equilibrium, all players correctly estimate the expected payoffs for each of their pure strategies, and play according to their noisy best-response function. The payoff estimates reflect the actual mixed strategies played by opponents, including the effects of noise terms. The basic quantal-response model does not specify a particular distribution for the noise terms. The most common form in the literature is *logit equilibrium*, in which the noise terms η_{s_i} are drawn independently from an extreme value distribution with cumulative distribution $F_i(\eta_{s_i}) = e^{-e^{-\lambda \cdot \eta_{s_i}}}$. For this distribution of noise terms, the choice probabilities for players' strategies take the form of a logistic choice function:

$$\Pr(s_i) = \frac{e^{\lambda \cdot u_i(s_i,\sigma_{-i})}}{\sum_{s_i' \in S_i} e^{\lambda \cdot u_i(s_i',\sigma_{-i})}}. \tag{4.3}$$

The logit equilibrium was first studied by McKelvey and Palfrey (1995), and an implementation of this model is available in the Gambit software package (Turocy, 2005; McKelvey et al., 2006). I use this implementation as the basis of the *Logit Equilibrium Solver* (LES), which solves for a logit equilibrium of the observed game and plays a best response to this equilibrium prediction. The parameter λ of the logit model controls the magnitude of the noise terms in the best-response functions, and the corresponding distribution of choice probabilities. In the limit as $\lambda \rightarrow 0$, the noise terms become infinite and equilibrium play approaches the uniform random distribution over the pure-strategy outcomes. As $\lambda \rightarrow \infty$, the noise terms approach zero and the logit equilibrium converges to a Nash equilibrium. The Gambit logit equilibrium solver uses a tracing procedure to compute equilibria for different settings of λ. Essentially, the algorithm starts from the centroid and traces the solution correspondence for small changes of λ. For very large values of λ, the solution correspondence converges to a Nash equilibrium for (almost all) games.[5] By solving for logit equilibrium with large values of λ, this solver can also be used to closely approximate sample mixed-strategy Nash equilibria. Logit equilibrium is particularly interesting as a meta-strategy for two reasons. It was originally developed as a model of human behavior in games, and has been used to explain various phenomena that have been observed

[5] The equilibrium is a unique selection from the set of equilibria, because the algorithm converges to the same equilibrium if executed multiple times.

in laboratory experiments. I return to this point following presentation of the results. Logit equilibrium is also an exact Bayes-Nash equilibrium of an incomplete-information game in which players have perturbed payoffs. In this game, players have a common observation of a base game, but privately observe a vector of payoffs for playing each of their pure strategies. Players act to maximize the combination of their payoff in the game and the payoff they receive for the pure strategy. Each player knows their own payoffs with certainty, but does not observe the perturbations for the other players' payoffs. A quantal-response equilibrium with the appropriate distribution of noise terms is a Bayes-Nash equilibrium of this game.

This setup has many similarities to a meta-game, and could potentially be modeled as a meta-game with an appropriate observation model. However, it makes some simplifying assumptions that do not hold in the general meta-game model or my experimental setup. First, players know their own payoffs with certainty, and are only uncertain about others' payoffs. Second, the noise terms are associated with pure strategies, rather than strategy profiles. This assumes that the noise is independent of the other players' strategy choices, which does not hold for estimated game matrices. To estimate a game matrix, payoffs are usually sampled by running simulations for a particular profile of strategies. More simulations may be run for some profiles that others, so the amount of uncertainty for the payoffs associated with different strategy profiles may vary substantially. Finally, in the specific case of logit equilibrium (which the available solver implements), the distribution of noise is restricted to the extreme value distribution, which may not be a good model of the noise in an estimated payoff matrix. Nevertheless, it is quite conceivable that the coarser noise model of logit equilibria is a good approximation in many cases.

4.2 Game Classes and Observation Models

The experiments presented in this chapter explore three base game classes paired with two distinct modes of observation. While clearly not exhaustive, these variations represent a diverse set of plausible conditions. All of the games are 2-player, 4-strategy non-symmetric games with payoff normalized to the range $[0,1]$. The three classes of games are *uniform random*, *common interest*, and *constant sum*. I generate instances from these classes using the GAMUT tool developed by Nudelman et al. (2004). Uniform random games are instances of the GAMUT *random game* class, while the common interest and constant sum game classes are both instances of *covariance games*, with covariance of 1 and –1, respectively. Covariance games are generated by drawing the payoffs for each outcome profile from a multivariate normal distribution, fixing the correlation between each pair of players' payoffs. I generated 10,000 sample instances of games from each of these three classes, and used the same instances for all experiments in this chapter.

I consider two observation models: *stochastic observation* and *incomplete observation*. The stochastic observation model generates empirical games by adding independent mean-zero Gaussian noise to each payoff in the base game. I vary the level of uncertainty in these observations by varying the standard deviation of the distribution. The incomplete observation model generates empirical games by revealing the exact payoffs for a random subset of the pure-strategy profiles. No additional information about the payoffs for the remaining profiles is revealed; only the mean payoff for the game class is available as an estimate.[6] The level of uncertainty in these observations depends on the number of profiles for which payoffs are revealed.

These two observation models correspond to different types of uncertainty common in empirical game-theoretic analysis. The stochastic observation model corresponds roughly to the presence of sampling noise due to stochastic elements in simulation data. In this model, players have the same amount of information about the payoffs for each strategy profile. The incomplete model corresponds to case where there is no direct information available about some strategy profiles. For example, the players may have a reliable source of payoff information, but only a limited search capability. In this model, the amount of information

[6] The mean estimate is 0.5 for all classes defined here.

Name	Parameters	Prediction Method
BRU	None	Uniform random opponent play
ENS	τ	Form distribution over pure-strategy profiles, with lower-regret profiles more likely
RDS	δ	Approximate a sample mixed-strategy Nash equilibrium, weight equilibrium and uniform distributions according to δ
LES	λ	Compute logit equilibrium for given λ, assuming players use noisy best-response functions

Table 4.1 Summary of meta-strategy families and their associated free parameters.

players have about different parts of the profile space varies. Many real domains (such as the TAC SCM game in Chapter 2) contain both forms of uncertainty, but here I isolate the two cases for experimentation.

4.3 Noisy Observation and Distributional Predictions

The three families of meta-strategies defined in the previous section (ENS, RDS, and LES) are summarized in Table 4.1, along with the BRU meta-strategy defined in Section 3.3. All three families are based on the idea of generalizing equilibrium concepts for complete information games to make distributional predictions. These predictions are distributions over the pure-strategy profiles, and the meta-strategies select strategies by computing best-responses to these distributions. Each has a single parameter that interpolates between a uniform random prediction and single equilibrium strategy profile, defined by the associated solution concept. The algorithms differ in which equilibrium concept is used and how the interpolation is performed. An example of how the prediction changes for different parameter setting is shown in Figure 4.1. The figure presents a visualization of how the predictions of the LES meta-strategy change for different settings of the parameter λ (see Section 4.1.3). All predictions shown are for the same payoff matrix. At one end of the parameter space, the predictions are very flat, and all profiles are almost equally likely. At the other, the probability mass is concentrated on the profiles that are closest to equilibrium.

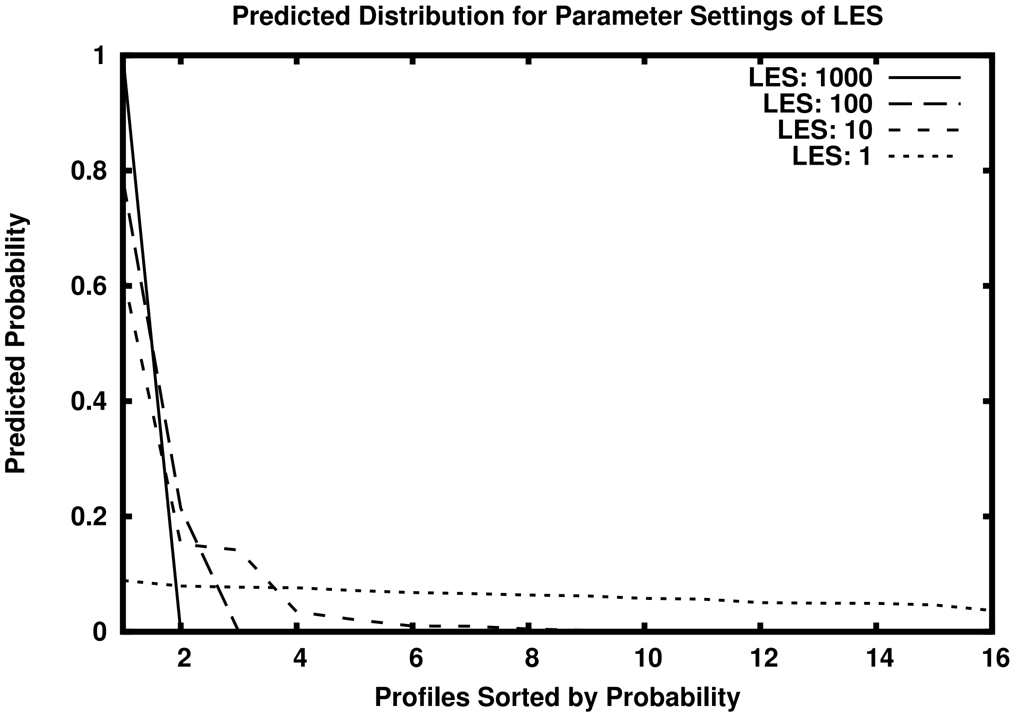


Figure 4.1 How changes in the parameter setting affect the LES predicted distribution over profiles. As the parameter λ is decreased, the predicted outcome distributes weight more evenly across the pure-strategy profiles. Each curve represents a distribution. The profiles are sorted by probability, so the profile with the highest probability appears on the far left.

The parameters of the other meta-strategies have qualitatively similar effects on predictions.

Observation uncertainty makes it more difficult to predict what other players will do, and distributional predictions are a way to represent this uncertainty. When players have good information about the game, they may be able to make very specific predictions about what others will do and, in turn, select a response with a very high payoff in this specific context. However, selecting a best-response based on a specific prediction also carries a risk if the prediction is inaccurate; the selected response may be arbitrarily bad if something unexpected occurs. Predicting a broader distribution tends to encourage less risky actions, but may forgo higher payoffs. The hypothesis is that the parameterized distributional predic-

Meta-Strategy Family	Candidate Parameter Settings
ENS (ε)	0.01, 0.05, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5, 0.75, 1.0
RDS (δ)	0, 0.11, 0.22, 0.33, 0.44, 0.55, 0.66, 0.77, 0.88, 0.99
LES (λ)	1, 5, 10, 25, 50, 75, 100, 250, 500, 1000

Table 4.2 Meta-strategy parameter settings used in the experimental setup.

tions made by ENS, RDS, and LES capture this tradeoff, and allow for a useful spectrum of predictions depending on the quality of observations available. If this is the case, we should observe a regular pattern in the preferred parameter settings as observation noise is varied.

I set up an experiment to test the relative performance of different parameter settings as observation noise is varied. For each meta-strategy family, I selected 10 candidate parameter settings, which are listed Table 4.2. These are distributed across the interesting range for each family, with parameter settings at one extreme yielding predictions that are close to uniform random, and setting at the other end yielding predictions close to equilibrium. In this experiment, I test the parameter settings only against other members of meta-strategies; each parameter setting is a distinct meta-strategy.

To compare the performance of the parameter settings, I define a performance metric based on stability in the meta-game. The *homogeneous regret* for each meta-strategy (i.e., parameter setting) is the regret ε for the profile in which all players play according to the meta-strategy. Better performance on this measure is defined by lower regret. A lower regret means that there is less benefit (or higher loss) associated with switching the best alternative meta-strategy, within the constrained space of candidate meta-strategies (which for this experiment consists of different parameter settings for the same family).[7] A desirable features of this measure is that it identifies meta-strategies that form equilibria of the meta-game in self-play. In most cases, these homogeneous profiles are among the most stable profiles that any given meta-strategy participates in, so this measure also provides a rough indication of

[7] No single-dimensional measure of performance can accurately summarize the performance of a strategy in an arbitrary game. I experimented with several other measures, all of which support the results presented here.

the best-case stability for each meta-strategy.

I present results for the uniform random class of games, using both the stochastic and incomplete observation models.[8] For each observation model, I simulate the meta-games for a range of different noise levels. Each meta-game (corresponding to a specific noise level) is estimated based on 10,000 sample base games, with one observation of each game. The regret measure is computed from the average benefit to deviating over these sample games.[9]

The plots in Figures 4.2, 4.3, and 4.4 show the parameter setting with the best performance as observation noise is varied, where performance is defined by the homogeneous regret measure. Two plots are shown for each family of meta-strategies, corresponding to the qualitatively different observation models. A general pattern is clear in data: as observation noise is increased, the meta-strategies with lowest homogeneous regret are those which make broader predictions about opponent play, distributing weight more evenly over the possible strategy profiles. Though the data is not presented here, this general pattern also holds for the other two classes of games described in Section 4.2. This appears to be a robust phenomenon, and supports the general intuition that solution methods should make less specific predictions about the outcome of a game when there is uncertainty about the game model. For the specific meta-strategies tested, this evidence supports the claim that these parameterizations capture (approximately) the effects of observation noise on opponent play, and do so in a way that improves strategy selection.

4.4 Benchmarking the Meta-Strategies

In this section I present direct comparisons between the families of meta-strategies. I test the candidates using several combinations of different game classes and observation models. Results are presented using the same homogeneous regret metric as the previous experiment,

[8] I ran the same experiment on the other two game classes and observe an identical qualitative result.

[9] Given the large sample size, the confidence intervals for the regret measures are very small (on the order of 10^{-4}). However, in some cases the difference in regret for neighboring parameter settings is also very small, and the parameter settings are almost equivalent.

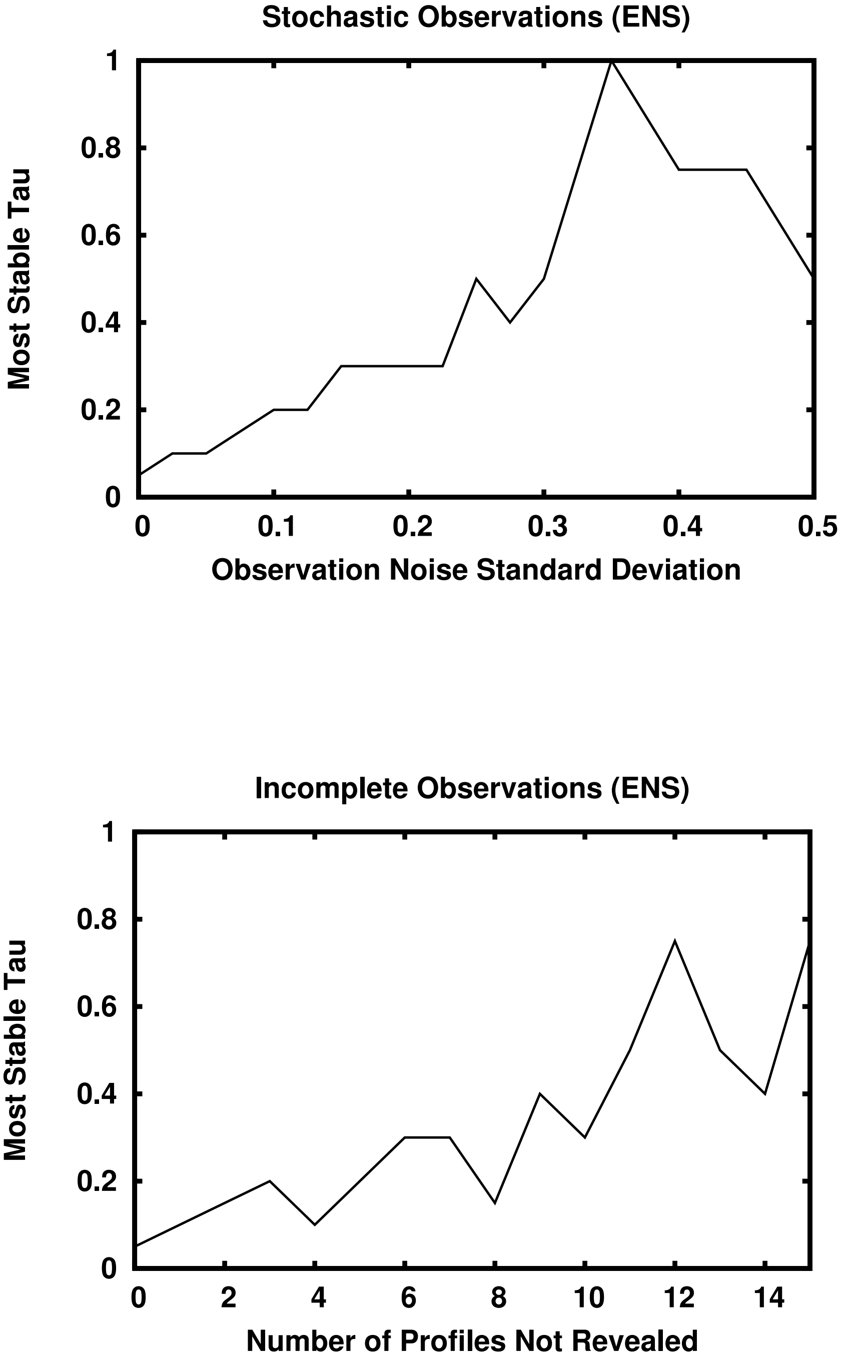


Figure 4.2 Most stable settings of the ENS parameter τ across different levels of noisy observations on random games. Results for the stochastic observation model are on top, and the incomplete observation model on the bottom. Higher settings of τ correspond to predictions that are closer to random play. Observation noise is increasing left to right.

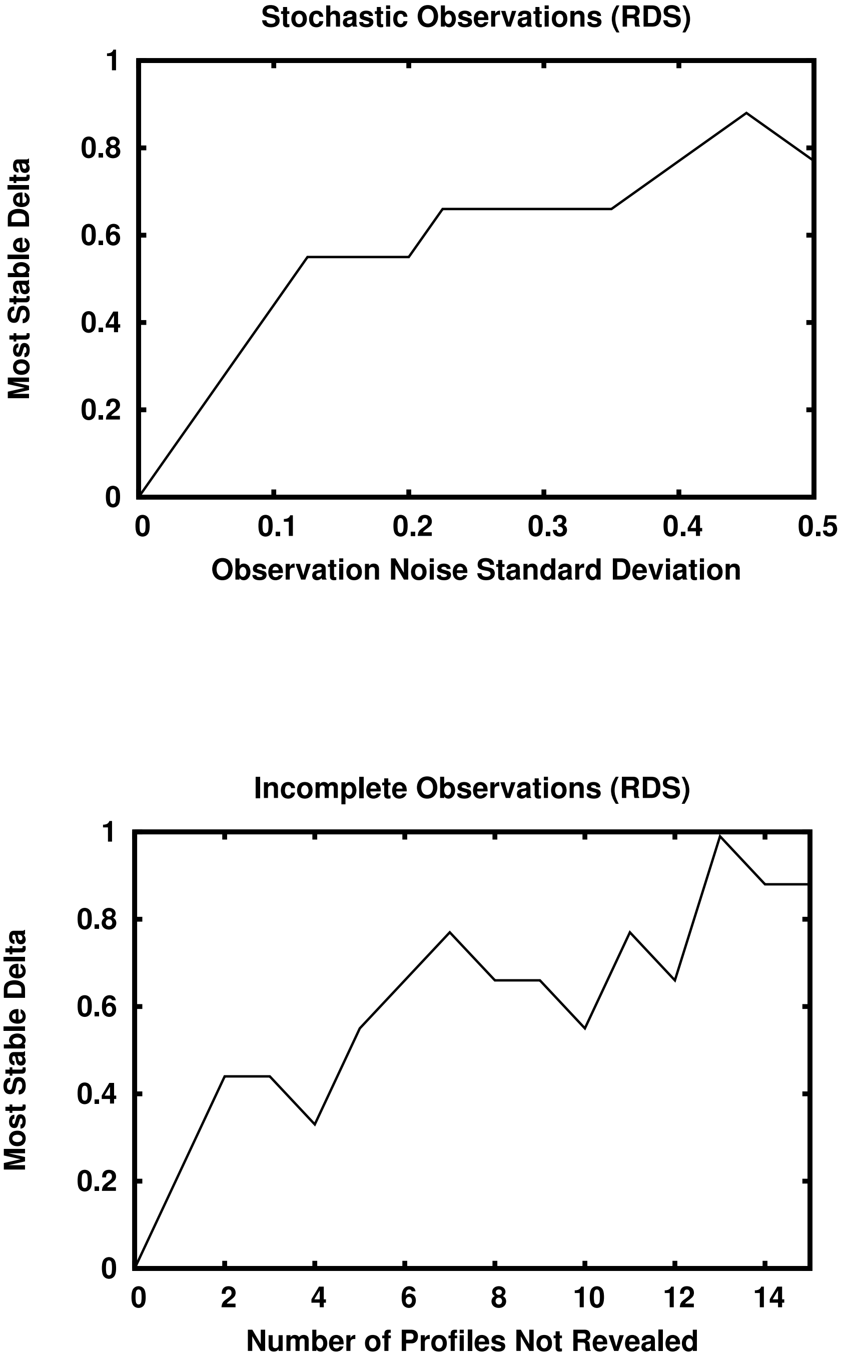
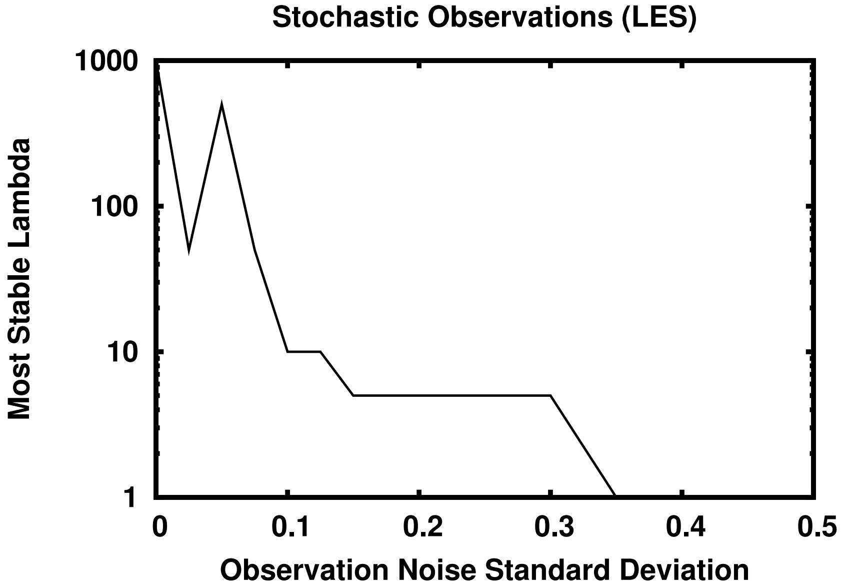


Figure 4.3 Most stable settings of the RDS parameter δ across different levels of noisy observations on random games. Results for the stochastic observation model are on top, and the incomplete observation model on the bottom. Higher settings of δ correspond to predictions that are closer to random play. Observation noise is increasing left to right.



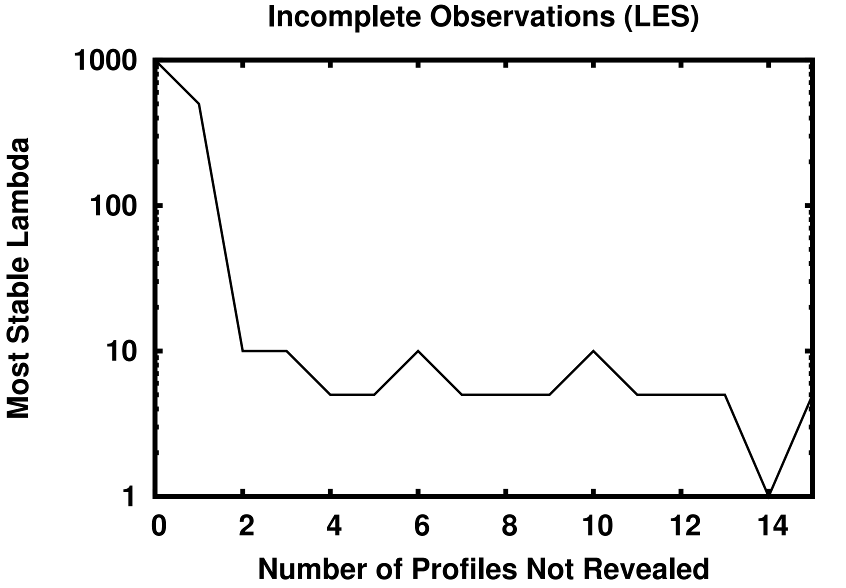


Figure 4.4 Most stable settings of the LES parameter λ across different levels of noisy observations on random games. Results for the stochastic observation model are on top, and the incomplete observation model on the bottom. Lower settings of λ correspond to predictions that are closer to random play (opposite of ENS and RDS). Observation noise is increasing left to right.

defined in Section 4.3. For each candidate meta-strategy, performance is measured as the regret for the homogeneous profile where all players use the same meta-strategy. The experiments test a single instance of each family of meta-strategies, with the addition of BRU as a benchmark (defined in Section 3.3). I set the parameters for the meta-strategies using the results of the previous section, which tested 10 possible parameter settings for each meta-strategy in a self-play setting. The parameter settings for each meta-strategy family vary according to the context, including the game class, observation model, and noise level. The same procedure is used to set parameters for all three families, selecting the setting with minimum homogeneous regret in self-play for the same context.

There are two reasons to restrict the experiments to a single parameter setting for each meta-strategy family. One is to keep the computational cost for simulating and analyzing the meta-games manageable. The other is that the regret measure presented is based on deviations between different meta-strategies. If the candidates include more than one parameter setting for a meta-strategy family, the meta-strategies from the same family often exhibit very similar performance. This affects the measured regret in cases where the best alternative is a different parameter setting from the same family. Some interesting features of the data can be masked by this, particularly in cases where there is a large loss for deviating to another family of meta-strategies, but a small loss for deviating to a different parameter setting within the same family.[10]

My experiments consider six categories of meta-games that combine the three game classes and two observation models from Section 4.2. Within each of these categories, I vary the level of uncertainty as in previous experiments, and the meta-strategy parameter settings as previously described. For each category and level of uncertainty, I estimate the

[10] It is possible to run similar experiments using multiple instances of each family with different parameter settings, but the results are more difficult to parse. The first issue is the effect of including similar meta-strategies on the regret measure. This causes regrets to cluster around zero rather than go into negative territory, losing some granularity for low-regret meta-strategies (positive regrets are still observed). The other is that plots showing more than a few meta-strategies become increasingly difficult to read. In preliminary tests, I did not find any instances where including additional meta-strategies from the same family led to qualitatively different conclusions about the relative performance of the families.

meta-game using 10,000 sample base game instances, which a single observation of each base game. The results are presented in Figures 4.5, 4.6, and 4.7, showing performance on the homogeneous regret measure for the three meta-strategy families and the BRU baseline.[11] On this measure, lower values indicate stronger performance. Any value less than 0 is particularly interesting, as it indicates a homogeneous profile that is a Nash equilibrium of the constrained meta-game; in other words, there are no beneficial deviations from this meta-strategy if all players use it. Negative values on this measure indicate the size of the loss that players would expect for using the next-best alternative among the remaining candidates.

The first interesting feature of the experimental data is that homogeneous profiles of ENS, RDS, and LES are all meta-strategy equilibria for low noise levels in both the random and common interest game classes. This is not especially surprising since all three methods play according to approximate equilibria of the base game in this case. More interesting is that deviations between families often result in significant net losses. This is evidence of a coordination effect. Players using the same meta-strategy tend to coordinate on the same equilibrium of the complete information game when there is low noise, but different families of meta-strategies tend to select different equilibria. The differences among the meta-strategies are generally greatest at low- and mid-range noise levels, indicated by larger regret magnitudes in these ranges. As observations become very noisy, the regret for the meta-strategies generally converges to zero (with the exception of BRU for the incomplete observation model). This is also intuitive; under very noisy conditions, the outcome of any analysis depends largely on the noise and the final selections are close to random for all meta-strategies.

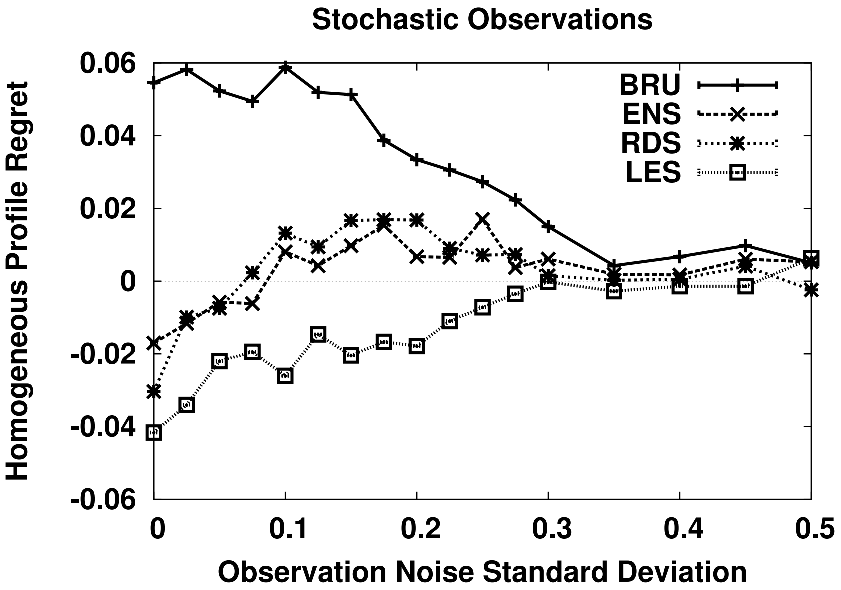
The most striking result is the strong performance of the LES family of meta-strategies

[11] The 95% confidence intervals are shown in the plots, but are too small to distinguish in most cases, since the intervals are roughly 10^{-4}. The intervals are computed by finding the benefit to deviating between the relevant pair of meta-strategies in each individual base game sample, and taking the variance of these data points. Most of the variation observed in the data points comes not from sampling noise associated with the base game and observations, but from the effects of varying the parameter settings fro each meta-strategy in the different contexts.

across the range of tested conditions. The homogeneous profile of this meta-strategy is almost always an equilibrium of the constrained meta-game. In many cases, it is the only equilibrium among the homogeneous profiles. When there are multiple equilibria, LES often forms the equilibria with maximal loss for deviations. The performance of LES relative to the other meta-strategies is especially strong when there is moderate uncertainty. In all six combinations of game class and observation model, LES forms the *only* homogeneous equilibrium in (roughly) the middle third of the range for observation uncertainty. In this range the meta-strategies typically have large incentives to deviate to LES.

There is one exception where there are significant incentives to deviate from the homogeneous LES profile. This occurs for low noise levels in constant sum games; in these cases, RDS forms an equilibrium. A distinctive feature of this class of games is that the instances frequently possess only mixed-strategy equilibria. Additional analysis of this case reveals that this is to some extent an artifact of computing a best-response to the predictions. The LES meta-strategy identifies very close approximations to equilibria of the compete information game when observations have little or no uncertainty. If the meta-strategy simply played the strategy specified in the equilibrium, there is no beneficial deviation of the magnitude observed in the data. However, the meta-strategy actually selects a strategy that is a best-response to the equilibrium. This computation selects the pure strategy with the maximum expected payoff, given the predicted distribution over opponent play.[12] There may be beneficial deviations from these pure strategies. This phenomenon suggests that there may be benefits to considering meta-strategies that use softmax or other methods to compute mixed-strategies in the best-response computation.

[12] In the case of ties, a uniform mixture over the pure strategies is used. However, this rarely happens due to numerical approximation—even in the case of mixed-strategy equilibria, when all pure strategies played with positive probability theoretically have the same expected payoffs.



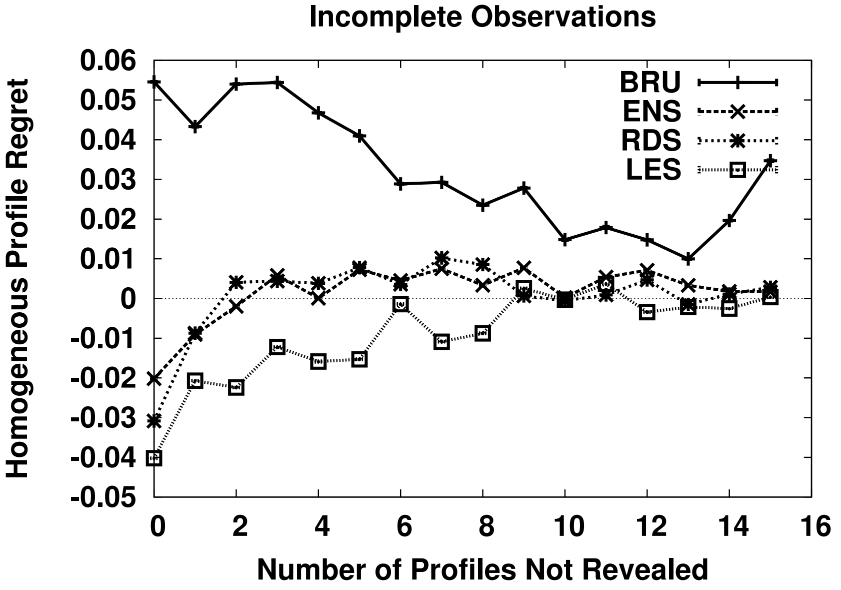
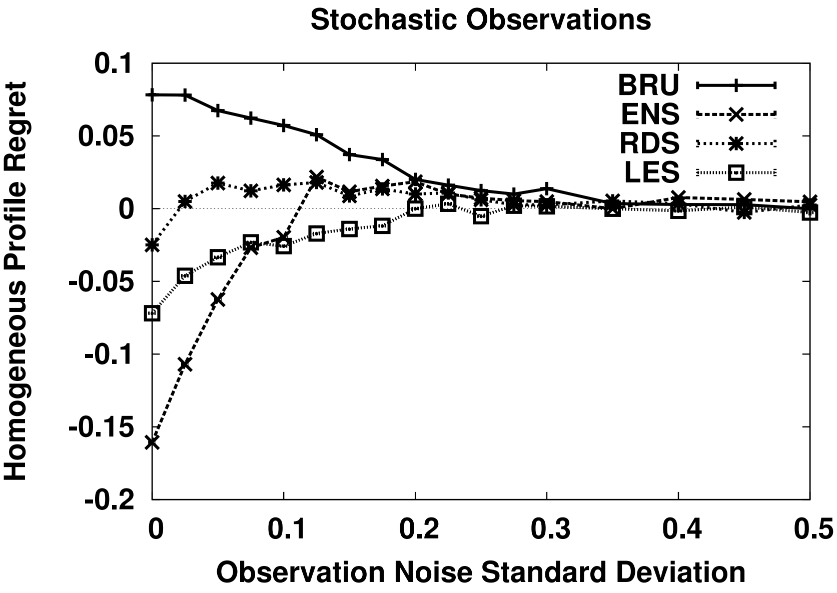


Figure 4.5 Uniform random games. Comparison of four meta-strategies playing uniform random games with stochastic observations (top) and incomplete observations (bottom).



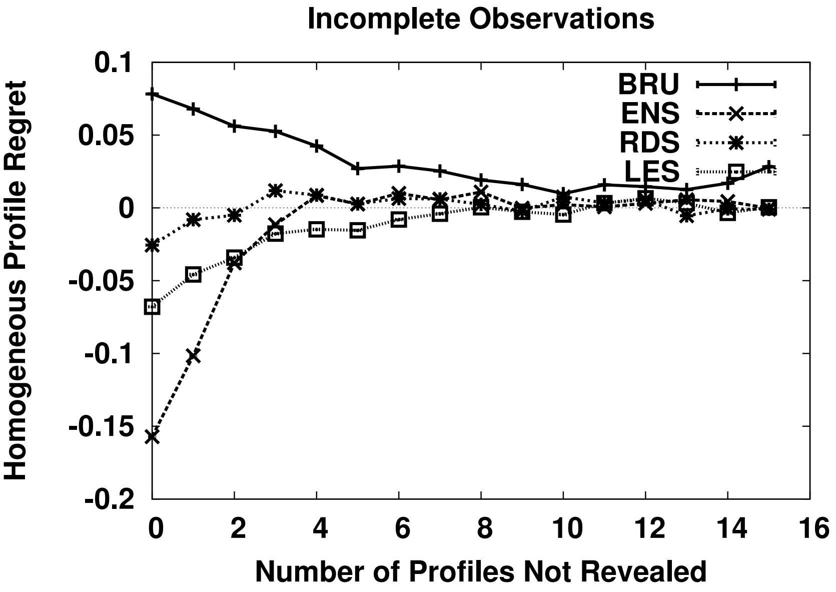
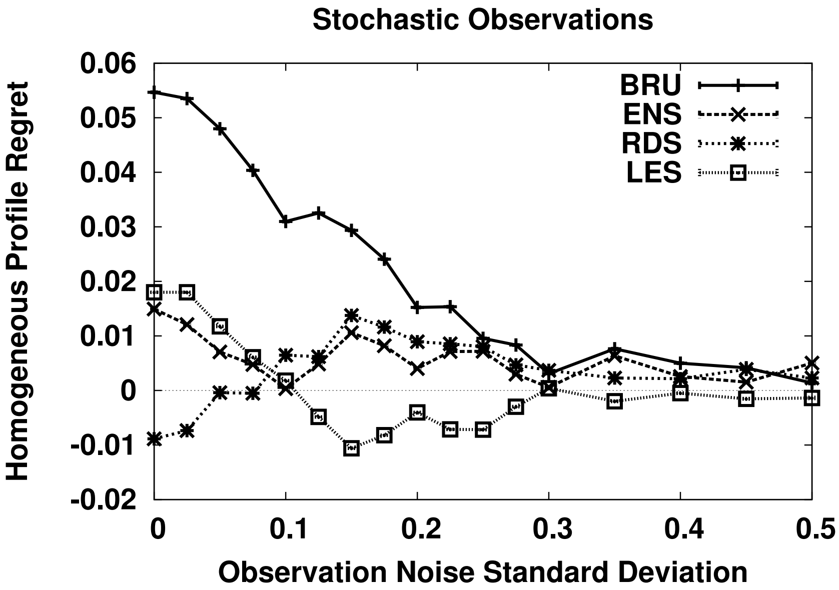


Figure 4.6 Common interest games. Comparison of four meta-strategies playing common interest games with stochastic observations (top) and incomplete observations (bottom).



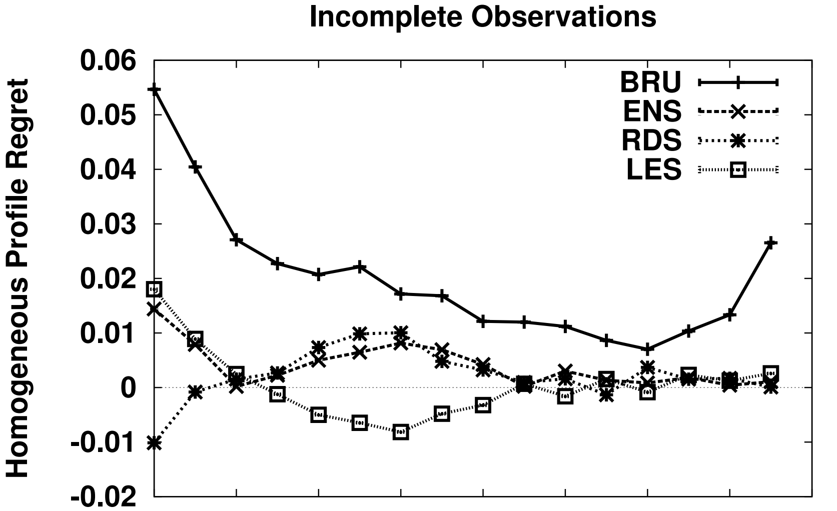


Figure 4.7 Constant sum games. Comparison of four meta-strategies playing constant sum games with stochastic observations (top) and incomplete observations (bottom).

4.5 Discussion

The logit equilibrium family of meta-strategies emerged as the clear champion in my experiments, with compelling performance across a range of meta-games. This result establishes a baseline against which future strategy selection methods may be tested. It also has immediate relevance for applications of empirical game-theoretic analysis, such as the one presented in Chapter 2. The results suggest that logit equilibrium should play a central role in analyzing the estimated games produced in these applications, especially if strategy selection is the primary objective. For practical reasons, it is especially encouraging that meta-strategies from the LES family performed well across varying conditions. The high-level approach used by LES to factor noise into the equilibrium predictions generalizes to give good approximations for the range of game classes and observation models considered. In my experiments, I used initial evidence to tune the parameter of this method to the specific meta-games.[13] The data contain some indications that there is hope for generalizing even these parameter settings using gross characterizations of the model. In Figure 4.4, the pattern of parameter settings selected is remarkably similar across the two observation models, varying in a regular way with the level of uncertainty in the observations.

Logit equilibrium is also of broader interest because it explains many qualitative aspects of human behavior that are not predicted by Nash equilibrium (McKelvey and Palfrey, 1995; Anderson et al., 2001; Capra et al., 1999). My results support logit equilibrium based on a different set of motivations, and provide a new perspective on the success of this solution concept in the behavioral setting. Humans participate in frequent interactions with others, and many of these have relatively low stakes. Under these conditions, it is quite reasonable to expect humans to adopt general methods for strategic reasoning that can be easily adapted to different contexts. Uncertainty is also ubiquitous in these interactions, and may be ascribed at least in part to payoff uncertainty. My results provide some justification

[13] Applications of logit equilibrium to analyze empirical data typically use maximum likelihood methods to estimate the parameter λ (McKelvey and Palfrey, 1995; Capra et al., 1999).

for logit equilibrium under similar conditions, where uncertainty is a principal feature of the strategic setting and general methods are preferable.

Another principle supported by my experimental results is that broader predictions of play are preferable to narrow predictions when there is uncertainty about the game. This is particularly relevant to the practice of applying complete information equilibrium concepts to estimated game models. My results show that these predictions are not robust to the presence of substantive uncertainty (see Section 3.3 for the most direct evidence for this point). In realistic applications of game theory where uncertainty and ambiguity are the norm, strategic reasoning that relies too heavily on strong informational assumptions is unlikely to yield desirable results. This is also relevant to the literature on equilibrium refinements that seek to narrow the predictions of the Nash equilibrium model, ideally to a unique prediction. While such definitive predictions are satisfying from a theoretical perspective, new solution concepts that broaden the predictions of the model may be more useful in practice. I investigate strategy selection methods that make broader predictions by generalizing complete information equilibrium concepts. These methods retain many of the essential features of the strategic reasoning employed by these concepts, but offer better performance under uncertainty.

4.6 Related Work

Many works in game theory have considered the question of how payoff uncertainty affects game-theoretic analysis. In one of the first theoretical applications, Harsanyi (1973) showed that small payoff perturbations can motivate mixed strategies. Payoff noise has also been used to motivate many of the equilibrium refinements introduced in the literature, including trembling-hand perfection (Selten, 1975) and proper equilibrium (Myerson, 1978). Fudenberg et al. (1988) and Dekel and Fudenberg (1990) investigate the robustness of solution concepts for complete information games to the introduce of small amounts of payoff noise.

In global games (Carlsson and van Damme, 1993), the lack of common knowledge resulting from payoff noise can lead to a unique equilibrium selection. This was shown first for 2-player, 2-action games, but later extended more general settings (Frankel et al., 2003). Similar results showing that noise can lead to the selection of particular equilibria have also been derived for evolutionary models (Foster and Young, 1990; Kandori et al., 1993; Blume, 2003).

A common feature of these models is that they derive results for limiting cases with very small payoff perturbations. My interest here is in cases where noise terms do not vanish, but play a significant role in the analysis. In addition to quantal-response equilibrium (which was a candidate meta-strategy), there are several other models in which noise does not vanish in the limit. An earlier example is imperfect equilibrium, in which players attempt to implement a target strategy, but make mistakes in doing so (Beja, 1992). More recently, Goeree and Holt (2004) have developed a model of noisy introspection that has many similarities to quantal-response equilibrium. The cognitive hierarchies model (Camerer et al., 2004) focuses on the reasoning used by players, rather than payoff uncertainty. However, the resulting solution method shares some common features with behavioral models based on payoff uncertainty.

The meta-strategies I study in this chapter seek to maximize payoffs in the expected case by making distributional predictions. There are other approaches to dealing with uncertainty in games, including worst-case analysis and set-based solution concepts. Tennenholtz (2002) proposes competitive safety analysis, which seeks to guarantee agents a large fraction of the payoff they could receive by playing an equilibrium strategy (in some cases, the entire value). The robust game theory paradigm introduced Aghassi and Bertsimas (2006) deals specifically with payoff uncertainty, with the objective of maximizing performance for the worst-case realization of payoffs. Another broad approach to making broader predictions about game play is predicting sets of possible solutions. Two solution concepts that take this approach are rationalizable strategies (Bernheim, 1984; Pearce, 1984) and closed under rational behavior (CURB) subsets (Basu and Weibull, 1991).

Modeling the behavior of human players in an important goal for game theory. One of the interesting aspects of my results is that the logit equilibrium methods I find most effective for playing meta-games is also one that has achieved significant success in explaining behavioral results. One possible reason for this is that humans behave as if they are uncertain about the payoffs, even in very controlled laboratory conditions. This is quite plausible, as it may be difficult to induce certainty about the game, even in idealized settings (Weibull, 2004). There have been many empirical studies that evaluate how closely theoretical models fit the data generated by human subjects in laboratory settings (Erev and Roth, 1998; Goeree and Holt, 2001; Friedman, 1996; Hopkins, 2002; McKelvey and Palfrey, 1995). Several focus explicitly on how well humans are able to learn or play games in limited information settings (Chen and Khoroshilov, 2003; Meyer and Roth, 2006; Oechssler and Schipper, 2003). My meta-game analysis offers a evidence of a different sort, evaluating how well proposed algorithms (e.g., logit equilibrium) perform in normative settings. Haile et al. (2008) raises specific concerns about the testability of the quantal-response equilibrium model by showing that the basic model places no testable restrictions on the empirical data that can be observed from a single sample game, due to the power of the noise terms in fitting the data. While the logit equilibrium model is more restrictive and does make testable predictions, this is a legitimate concern for testing complex behavioral models using behavioral data. My experimental framework is particularly suited to running tests across many game instances, offering a complementary means of testing behavioral models that avoids this problem to some degree.

Chapter 5

Exploration Policies for Large Games

For games with large strategy spaces, exhaustive analysis based on enumerating the payoffs for all strategy profiles is often infeasible. This is one of the key reasons that classic games such as chess are challenging to solve. It is not difficult to determine the payoffs for any specific instance of game play, given the sequence of moves. However, enumerating all of the possible paths of game play is well beyond the capabilities of any modern computer—even the specially-designed supercomputer Deep Blue (Campbell et al., 2002).

In this chapter I consider the setting where players can observe the payoffs for only a limited subset of pure-strategy profiles. This is similar to the incomplete observation model investigated in Chapter 4, except that players have control over the profiles they observe. Jordan et al. (2008) refer to this as the *reveal-payoff* model of observation, and I adopt this terminology for the remainder of the chapter. A player either knows all of the payoffs for a strategy profile with certainty, or has no (direct) information about the payoffs. Players may execute sequential queries to *evaluate* the payoffs for a pure-strategy profile. Limits on the amount of information players receive are modeled as bounds on the number of queries allowed. In general, each choice of which profile to evaluate may depend on the history of payoffs already revealed. Section 3.1.3 contains additional discussion of this mode of directed observation in the context of the meta-game model.

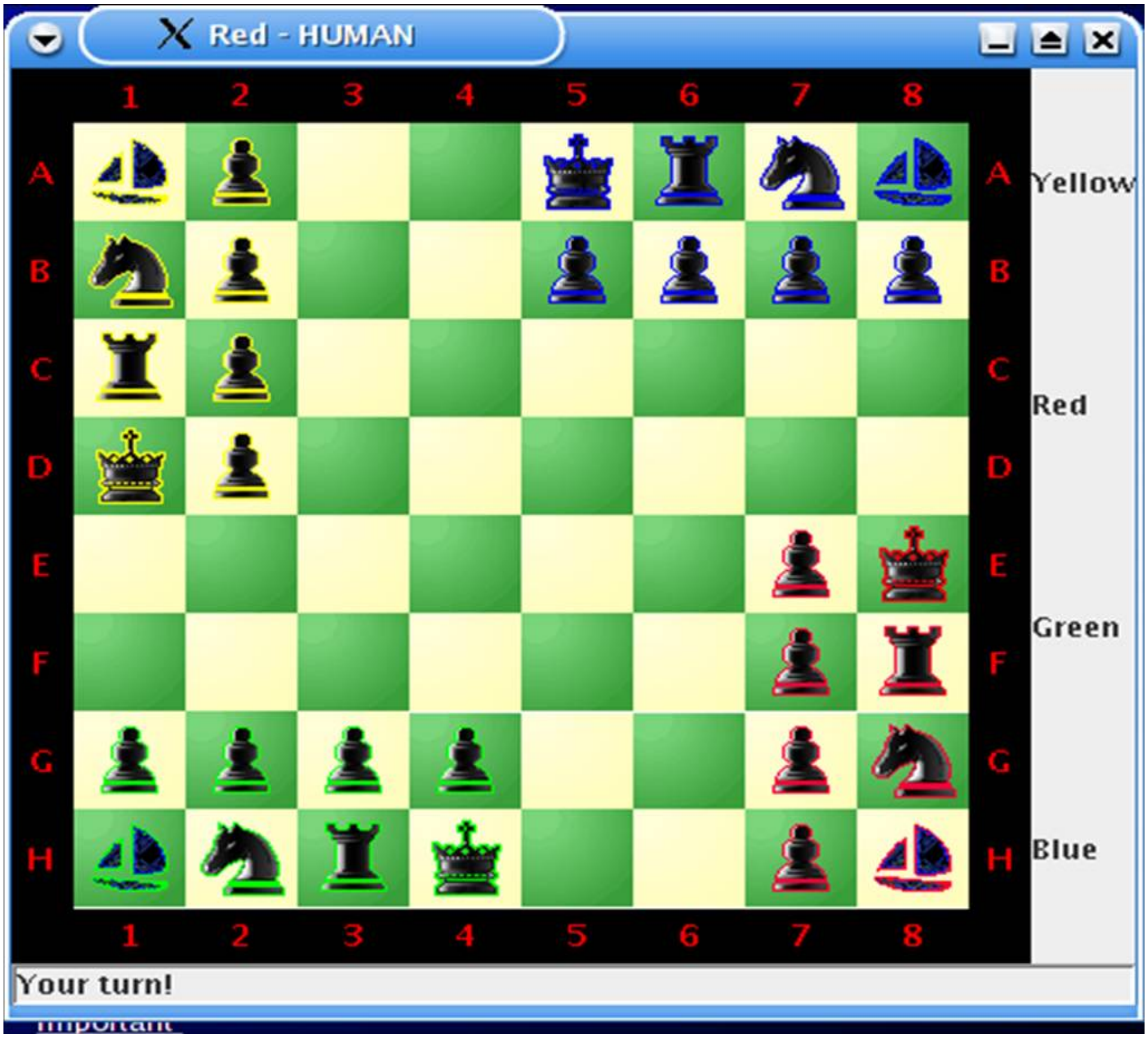
An *exploration policy* describes how a player chooses which profiles to evaluate. The central question of this chapter is to identify good exploration policies that yield valuable observations. In general, the quality of the observation will depend on the task for which the information is used. Here I consider both an equilibrium confirmation task and the strategy selection task from the previous chapter. An equilibrium is said to be *confirmed* when all deviations have been tested and no beneficial deviations have been found. One of the keys to a good exploration policy is exploiting structural regularities in the game. I begin with a case study of chaturanga (a four-player variant of chess) that provides evidence of structure in this game, and proposes some ways to exploit it in this domain. I test these intuitions in a more abstract setting, evaluating candidate exploration policies on broad classes of games with known structural properties. The candidates are all able to exploit various types of structure, both to find equilibria quickly and improve strategy selection.

5.1 Chaturanga

Chaturanga is a variant of chess for four players. The starting configuration of the game board is shown in Figure 5.1. Each player controls eight pieces: one king, rook, knight, and boat, and four pawns. The king, rook, and knight all move in the same way as the standard chess pieces. The boat moves diagonally in any direction, but must jump over exactly one square. Pawns advance and capture in the standard way, based on the player's orientation on the board. They do not have an initial double move. Upon reaching the final square, they may promote to any other type of piece except king. There is no notion of check or checkmate;[1] players are eliminated only when their king is taken. The pieces of eliminated players remain on the board as obstacles, but cannot move. The game ends when a single king remains, an identical position occurs three times during the game, or each player has made at least 50 consecutive moves without moving a pawn or capturing a piece.[2] At the end of the game, payoffs are determined by dividing one point equally between all players

[1] Unlike standard chess, taking an opponent's king when possible might not be the best move, since the game continues against the remaining opponents.

[2] These stopping conditions guarantee that games are finite.



Your turn!

Figure 5.1 The game board for chaturanga.

with a king remaining; all other players receive a payoff of zero.

5.1.1 A Parameterized Strategy Space for Chaturanga

A strategy for chaturanga defines a move for each possible board configuration. The first challenge in designing a strategy for this domain is that the number of possible board configurations is extremely large. In his original paper, Shannon (1950) gives a rough estimate (not accounting for reachability) that chess has 10^{43} legal board positions.[3] This

[3] The exact number is very difficult to determine, and remains unknown.

estimate also holds for chaturanga, which is played on the same size board with the same number of pieces. A tabular representation that specifies a move for each of these positions is too large to even fit into the memory of a modern computer. Programs for playing chess and similar board games typically solve this representational difficulty using an evaluation function. This function defines a value for each possible board configuration—typically interpreted as an estimate of the probability that each player will win the game from the given configuration (Thrun, 1995). An evaluation function may be used in conjunction with game tree search, which looks ahead in the game by considering possible sequences of moves.

I define a parameter space for chaturanga which adopts these methods from chess-playing programs, using both an evaluation function and game-tree search. The evaluation function and search procedure are defined by a set of parameters. These parameters define a *transformed* strategy space for chaturanga; settings of these parameters are valid chaturanga strategies, but the parameter space may not include all strategies in the original strategy space. Strategies in this space can be represented compactly, and the space is more amenable to analysis than the original strategy space. However, this transformation may modify the objective of the analysis. For instance, rather than identify the best strategy globally, the goal may be to identify the best strategy within the parameter space.

Evaluation Function

The evaluation function assigns a value, $Score_i(C)$, for each of the four players, given any given a legal board position C. Ideally, these values are predictions of the expected payoffs for each player when play continues from the given position. The evaluations are based on primitive features computed for each board position, representing factors such as material value, threats to pieces, protected pieces, control of board squares, and mobility of the king. Many specific features are motivated by features commonly used in chess programs, such as KnightCap (Baxter et al., 2000). The features are first combined to form an estimate of the raw strength of each players' position, denoted $R_i(C)$. Each piece P has an associated value $V(P)$ and threat level $T(P)$, with the king K treated separately due to its special role. Intuitively, the threat level is the probability that the piece will be captured in the near future. Both $V(\cdot)$ and $T(\cdot)$ are in turn defined by weighted combinations of the more primitive board features. Each feature has a weight, which depends on the type of the piece. These weights are the primary parameters of the evaluation function; there are roughly 60 of these weights in total. The functional form for positional strength combines these elements:

$$R_i(C) = T(K)\left[\sum_{P \in \{pieces \setminus K\}} T(P)V(P) + V(K)\right]. \tag{5.1}$$

In standard chess, the game has only two players and the payoffs are zero-sum. Translating the relative position strengths into an outcome prediction is relatively straightforward in this case. A complication arises in chaturanga due to the number of players: there are three pairs of players, and the relative strengths of these players need not be weighted equally in the evaluation. There is generally a non-linear relationship between the relative strengths of the players and the probability of winning.[4] I use a sigmoid function to represent pairwise comparisons for all players i and j:

$$PW_{i,j}(C) = \frac{1}{1 + e^{-\kappa_{i,j} \cdot (R_i(C) - R_j(C)))}}. \tag{5.2}$$

The multiplier $\kappa_{i,j}$ can be used to change the weight placed on differences in strength between each pair of players in the final evaluation. This is also a parameter of the evaluation function, and plays a particularly important role in my experiments. The pairwise strengths are combined and normalized to give the final value for each player:

$$Score_i(C) = \prod_{j \in \{players \setminus i\}} PW_{i,j}. \tag{5.3}$$

[4] In standard chess, a difference in strength equivalent to one pawn typically equates to near-certain victory (Baxter et al., 2000).

Game-Tree Search

Games with sequences of deterministic moves (like chaturanga) can be represented as a game tree. Each node in the tree represents a legal sequence of moves, and the edges represent the individual moves. Minimax search is a standard technique for searching portions of the game tree during a game. Starting from the current state, the search expands possible paths of play, typically stopping before a terminal state in the game. Each level of the tree corresponds to the possible moves for one player. Leaf nodes in the game tree are evaluated using the evaluation function. The values in these leaf nodes are propagated back up to the root of the tree using backwards induction, selecting the best move for the current player at each level of the tree. Variations of minimax search are used by virtually all chess programs, often with sophisticated methods of search control. Exactly why exactly minimax search improves play is not fully understood (Lustrek et al., 2005), but it works extremely well in practice. The standard version of minimax search applies only to 2-player games, but an algorithm called max^N (Luckhardt and Irani, 1986) extends minimax to N-player games. Sturtevant (2003) has studied several variations of this algorithm, including different pruning methods.

My strategy space for chaturanga contains two version of max^N. The first uses depth-limited depth-first search that expands all paths of play to a fixed depth. The second uses beam search, which expands only a fixed number of nodes with the highest values at each level of the game tree. This allows a deeper search for a given number of node expansions, but ignores some lines of play. These search procedures are part of the parameter space for chaturanga, in addition to the parameters of the evaluation function.

5.1.2 Strategic Independence in Chaturanga

The parameterized strategy space for chaturanga provides a compact representation for strategies, but transforms a game with a finite strategy space into one with an infinite strategy space by introducing continuous parameters. This does nothing to mitigate the problem of analyzing a large space of strategic choices. One general approach to solving large problem instances is to exploit structural properties. For chaturanga, this means identifying likely structural properties of the parameterized strategy space, and finding methods capable of exploiting them for game analysis. A natural candidate is some form of independence relationship between choices in the game. Several existing representations for games capture forms of independence, including multi-agent influence diagrams (MAIDs) (Koller and Milch, 2003), graphical games (Kearns et al., 2001), and action-graph games (Jiang and Leyton-Brown, 2006). These representations are potentially smaller than normal-form representations of the game, and researchers have exploited these compact representations to derive more efficient solution algorithms (e.g., for finding Nash equilibria).

Separating Strategic Parameters from Independent Parameters

The form of structure chaturanga may have is unknown, so one approach is to hypothesize some form of structure and look for supportive evidence. One indication that there may be regularities in the strategy space for chaturanga comes from analogy with chess. Ranking systems are very popular for chess and other competitive games (Stefani, 1997). The existence of a good ranking implies at least a weak form of transitivity, in that higher-ranked players are expected to beat all players of lower rank. Chess players are typically ranked using a numerical score computed from the results of tournament play (Glickman, 1995).[5] In the statistical models that underlie the rankings, this score reflects strength of each player. The winner for any match is determined by comparing the relative strengths of the players, with some noise.

One intuitive explanation for the transitivity implied by these rankings is that strong players are good at optimizing decisions that do not depend (much) on their opponent. This notion of *strategic independence* can be formalized as follows.[6] Suppose each player's

[5] Humans may use different strategies in different games, but it seems likely that they at least choose similar strategies. If this is the case, ranking players and ranking strategies is (roughly) analogous.

[6] A similar notion is defined by Koller and Milch (2003) for MAIDs.

strategy space is defined by a vector of decision variables $S_i = (X_i^1, \ldots, X_i^l)$ (in the previously defined chaturanga strategy space, these correspond to settings of parameters). If $u_i(s) = u_i(s')$ for all pairs of pure-strategy profiles s and s' that differ only in the choices for the two decision variables X_i^a and X_j^b, then X_i^a is strategically independent of X_j^b. In other words, the choice of X_i^a for player i does not depend on player j's choice for the variable X_j^b, because the payoffs are the same for all cases. The relationship is not necessarily symmetric, as player j's payoffs are allowed to vary for different combination of these two choices. It is plausible that some parameters in the chaturanga strategy space could be optimized independently of other players' choices, at least approximately. Some moves, such as defending or moving a king in danger, are likely to be good tactical decisions against virtually any set of opponents. On the other hand, high-level choices such as which opponent to attack may depend on analogous choices made by others.

Separating parameters that are strategically independent from parameters that have strategic interactions could aid game-theoretic analysis, by focusing strategic reasoning on the smaller strategy space defined by the strategic parameters. As a proof of concept, I propose one possible separation of the strategy space for chaturanga into strategic and independent parameters. In Section 5.1.3, I discuss experimental evidence for this separation of the parameter space. I single out a single parameter for strategic analysis. This parameter manipulates the value of $\kappa_{i,j}$ in Equation (5.2). Different weights are assigned to each pair of players in the evaluation function, and these weights depend on the spatial orientation of the players. As the weight placed on an opponent increases, a player will behave more aggressively towards that opponent, favoring moves that maximize the differences in positional strength between the two players.

For easy visualization, I project the pairwise weights into a single dimension called the attack angle. Nine possible settings of the parameter are shown in Figure 5.2. If the angle points directly at one opponent, almost all of the weight is placed on differences with that opponent in the evaluation. If the angle lies between two opponents, the weight is distributed

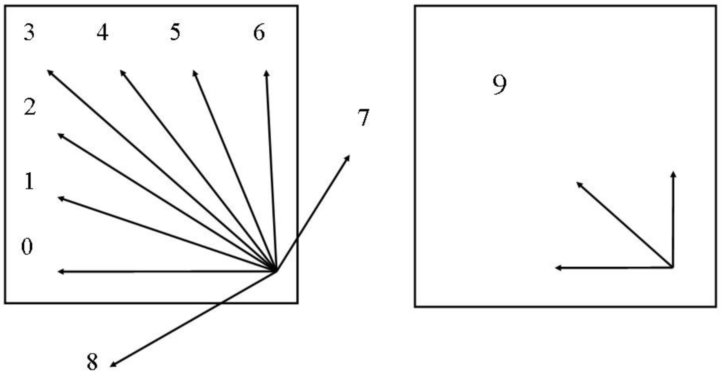


Figure 5.2 Nine settings of the attack angle parameter (left), and equal weighting of differences (right).

proportionally between those two opponents, ignoring the third. Also shown in the diagram is strategy 9, which weights all opponents equally. This strategy is the default weighting, but is not within the attack angle space.[7]

Local Search for Approximate Best-Response

If the parameterized strategy space I describe for chaturanga does have parameters that are strategically independent of others, how can this be exploited? Put another way, how should parameters that are strategically independent be optimized? Computing a best response to a fixed strategy profile is one possible approach. This computation only needs to test the possible deviations from the fixed profile, so it does not require knowledge of the payoffs for the full profile space. A best response selects the optimal value for all parameters, in the context of specific opponent strategies. The key observation is that the optimal setting for a parameter that is strategically independent of some opponent choices is *also* a best response to many other profiles that have not been considered. If the parameter is independent of all

[7] The attack angle parameter can specify weights on only two players, due to the projection into one dimension.

opponent choices, then the setting is already a global optimum after a single best-response computation. Best-response computations can be applied iteratively, in a process known as best-response dynamics (Fudenberg and Levine, 1998). In several classes of structured games, best-response dynamics is known to converge to Nash equilibrium, including supermodular games (Milgrom and Roberts, 1990) and potential games (Monderer and Shapley, 1996).

The parameterized chaturanga strategy space is infinite, so it is not feasible to compute exact best-responses. However, local search methods can approximate best-responses to a fixed strategy profile. Reinforcement learning is one technique that could be used for local search. There is a long history of using reinforcement learning to approximate evaluation functions for similar games: chess (Baxter et al., 2000), checkers (Samuel, 1959), and go (Schraudolph et al., 1994). The learned evaluation function depends on the opponents in the data set. If the set of opponents is fixed, learning will approximate a best response to the fixed set of opponents.

A common approach in practice is to learn using self-play as a training procedure. All players use the same evaluation function to play a game, and then update their evaluation functions using the learning rule before playing the next game. Learning in self-play is very much like best-response dynamics. Each learning update approximates a best response to the current profile. The profile is updated to reflect these best responses by using the new evaluation functions for the next game. The primary difference between best-response dynamics and self-play learning is that learning does not compute an exact best response—only a rough approximation. For chaturanga, it is possible that reinforcement learning in self-play could eventually converge to a profile of equilibrium strategies within the parameter space, but this is not guaranteed.[8] However, even if it does not converge it could still be a useful tool for strategic reasoning. One possibility is to use this learning method to generate candidates for more intense game-theoretic scrutiny. These candidates would already have

[8] Best-response dynamics may not converge in arbitrary games, and reinforcement learning using function approximation may also fail to converge to an optimal response.

optimal settings for independent parameters, as described above.

I use the standard TD(λ) reinforcement learning algorithm (Sutton and Barto, 1998) to learn approximate evaluation functions for chaturanga from simulated game data. For each game played, there is a sequence of board configurations that lead to the final outcome. A learning update is performed for each configuration. The error is defined by the difference between the payoffs predicted for the configuration, and the payoffs at the end of the game. The update takes a step in the direction of the gradient that maximally reduces the error. I use numerical approximation to estimate the gradient, testing small changes in each parameter to determine the effect on the error. The magnitude of the update depends on three factors: the magnitude of the error, a learning rate parameter, and the number of steps the board configuration is end of the game. Error is discounted further from the end of the game, using the rate given by the parameter λ of the TD algorithm.

5.1.3 Experiments

I present experimental evidence collected for two sets of chaturanga strategies.[9] Each strategy is defined by a vector of parameters specifying the evaluation function and online search mechanism. I introduce a small amount of noise into the strategies by forcing them to select a random move 5% of the time. This introduces variation into the games played between any given set of strategies. For each set of strategies, I estimate the payoffs for profiles of these strategies using a simulator for playing chaturanga games.[10]

The first set of strategies varies only the attack angle parameter described above. The second set holds the attack angle fixed (to the equal weighting), and varies the other parameters of the evaluation function and online search. I consider the 9 attack angle settings shown in Figure 5.2, as well as the uniform weighting. They are labeled using the indices in the figure. For this 4-player, 10-strategy game, I gathered at least 50 samples of the payoffs

[9] The data presented were collected for exploratory purposes, and are not specifically designed to test the hypotheses studied. Nevertheless, the data do provide some relevant evidence.

[10] The simulator was developed by Matt Abrams of Cougaar Software.

Table 5.1 Results of regressions for a player's score and the attack angle weights placed on the player.

	R^2	Coefficient
Sum of weights from all opponents	0.36	–0.14
Weight from right player	0.25	–0.20
Weight from left player	0.14	–0.07
Weight from diagonal player	0.03	–0.15

Table 5.2 Pure-strategy profiles with minimum regret in the attack angle strategy space.

ε	Profile	Scores
0.04	[7,8,8,2]	[0.22,0.23,0.21,0.33]
0.05	[8,7,4,7]	[0.24,0.38,0.27,0.11]
0.05	[7,1,8,8]	[0.16,0.29,0.18,0.37]
0.06	[2,5,8,7]	[0.09,0.26,0.52,0.12]
0.06	[9,7,4,7]	[0.25,0.46,0.17,0.13]
0.06	[9,9,9,7]	[0.17,0.37,0.23,0.24]
0.06	[9,7,5,8]	[0.40,0.12,0.32,0.16]
0.06	[9,9,5,3]	[0.44,0.25,0.15,0.17]
0.06	[4,0,7,7]	[0.26,0.24,0.12,0.38]
0.06	[4,7,8,6]	[0.12,0.42,0.15,0.31]

for each of the 10000 pure-strategy profiles. To verify that the attack angle actually has an impact on the scores of the players, I ran several linear regressions, shown in Table 5.1. Recall the attack angle parameter manipulates the weight placed on each opponent by a player's evaluation function. The weights in the table are the weight place on each player by various opponents, and correspond roughly to the strength of the opponents' attacks. Players who are the focus of other players' attacks do tend to have lower scores on average. The most potent attacks come from the player to the right, who can employ advancing pawns most easily in the attack.

The 10 most stable pure-strategy profiles for the attack angle space are shown in Table 5.2. Additional statistics summarizing other aspects of the data for the attack angle space are presented in Table 5.3. There is no pure-strategy Nash equilibrium in this space, and a diverse set of strategies appear in these 10 profiles. This demonstrates that each player's choice of attack angle is *not* independent of other players' attack angles. If these choices

Table 5.3 Statistics summarizing the data for the attack angle strategy space. Score is the average score over all profiles. Dev Benefit is the average benefit for deviating from the strategy to any other strategy. % Positive Dev is the percentage of deviations from this strategy that result in a net gain. The top X% columns show how often (as a percentage) each strategy appears in the given fraction of the pure-strategy profiles with minimum regret.

Strategy	Score	Dev Benefit	% Positive Dev	top 5%	top 1%	top 0.1%
0	0.214	0.04	66.3	4.5	2.2	2.5
6	0.222	0.031	62.7	4.7	2.5	2.5
3	0.238	0.013	54.9	8.2	5.5	2.5
1	0.245	0.005	51.9	8.3	5.2	2.5
5	0.246	0.004	51.2	9.8	12.5	7.5
2	0.248	0.003	50.3	9.0	6.8	5.0
4	0.255	-0.006	47.0	11.8	12.0	10.0
7	0.263	-0.015	43.2	13.2	15.2	30.0
8	0.279	-0.032	36.4	14.3	16.2	20.0
9	0.289	-0.043	32.1	16.2	21.8	17.5

were independent, there would be a dominant strategy for each player. Nor is this simply a case where all strategies are essentially equivalent, since the regressions clearly indicate that these choices do have an impact on payoffs. The attack angle is at least one instance of a parameter with strategic implications in the parameterized chaturanga strategy space.

The second strategy space varies parameters of the evaluation function and the online search method. I refer to this as the *evaluation/search* strategy space. There are 17 strategies in this space, briefly described in Table 5.4. The strategies were not generated using one particular methodology, but are diverse exemplars of possible parameter settings. Some have hand-crafted evaluation functions, while others were learned using the reinforcement learning method described above. Several of the evaluation functions are paired with different online search methods; unless otherwise specified, all strategies evaluate only the states resulting from the next move. There are a total of $17^4 = 83521$ profiles in this strategy space, and I collected at least 30 payoff samples for 7863 of these profiles, approximately 9% of the total profile space.

There are two pure-strategy Nash equilibria within this strategy space, and two other profiles with low regret. These are listed in Table 5.5. All of the deviations have been

Table 5.4 Strategies in the evaluation/search strategy space. A search ply represents a single move by a single player.

Strategy	Description
0	**Random.** Selects uniformly at random from available moves.
1	**Hand-set 1.** Hand-set weights.
2	**Hand-set 2.** Hand-set weights.
3	**Material only.** Weights for material (pieces) only.
4	**Learn 1.** Learned (1000 games self-play)
5	**Learn 2.** Learned (3000 games self-play)
6	**Hand-set 3.** Hand-set weights.
7	**Learn 3.** Learned (2000 games self-play)
8	**Learn 4.** Learned (1240 games, fixed opponent profile)
9	**1 DFS.** 2-ply DFS using strategy 1 evaluation function
10	**1 Beam.** 5-ply beam search using strategy 1 evaluation function
11	**2 DFS.** 2-ply DFS using strategy 2 evaluation function
12	**2 Beam.** 5-ply beam search using strategy 2 evaluation function
13	**3 DFS.** 2-ply DFS using strategy 3 evaluation function
14	**3 Beam** 5-ply beam search using strategy 3 evaluation function
15	**7 DFS.** 2-ply DFS using strategy 7 evaluation function
16	**7 Beam** 5-ply beam search using strategy 7 evaluation function

Table 5.5 Pure-strategy profiles with minimum regret in the evaluation/search strategy space

ε	Untested Deviations	Profile	Scores
0	0	[15,16,16,16]	[0.15,0.25,0.30,0.30]
0	0	[16,16,16,15]	[0.23,0.15,0.37,0.25]
0.05	0	[15,15,16,16]	[0.33,0.20,0.23,0.23]
0.02	1	[16,16,16,16]	[0.13,0.30,0.33,0.23]

tested for these profiles, with the exception of one deviation from profile [16,16,16,16]. The profile [15,15,16,16] has a beneficial deviation to the equilibrium profile [15,16,16,16]. The beneficial deviation for the profile [16,16,16,16] is to the equilibrium profile [16,16,16,15]. There are no other candidate equilibria in the data set that have an ε bound of less than 0.05 and fewer than 44 untested deviations. Additional summary statistics are provide in Table 5.6.

By any measure, strategies 15 and 16 are clearly the strongest in the exploratory strategy space. They are the only ones to appear in the pure-strategy Nash equilibria, and have much higher average scores than all other strategies. These are the two strategies within this space

Table 5.6 Statistics summarizing the data for the evaluation/search strategy space. Score is the average score over all profiles. Dev Benefit is the average benefit for deviating from the strategy to any other strategy. % Positive Dev is the percentage of deviations from this strategy that result in a net gain. The top X% columns show how often (as a percentage) each strategy appears in the given fraction of the pure-strategy profiles with minimum regret.

Strategy	Score	Dev Benefit	% Positive Dev	top 5%	top 1%	top 0.1%
3	0.049	0.160	77.9	0.5	0	0
0	0.051	0.183	81.1	0.3	0	0
11	0.054	0.146	75.7	1.2	0.8	0
12	0.064	0.143	72.0	1.2	0	0
2	0.089	0.148	72.2	1.7	0.8	0
1	0.105	0.131	67.5	1.7	0.8	0
14	0.110	0.078	58.6	0.8	0	0
10	0.177	0.034	47.6	1.7	0.8	0
6	0.199	0.019	47.4	2.5	2.5	0
9	0.213	0.019	47.9	2.5	1.7	0
4	0.221	0.009	44.2	3.4	1.7	0
13	0.255	-0.011	39.6	3.5	4.2	0
8	0.280	-0.041	34.2	3.2	2.5	0
7	0.353	-0.110	27.9	3.7	3.3	0
5	0.383	-0.155	21.0	8.3	5.0	0
15	0.480	-0.266	7.4	27.7	22.5	16.7
16	0.481	-0.261	8.3	36.0	53.3	83.3

that incorporate both a learned evaluation function and online search. The strategies with learned evaluation functions did very well overall; the three versions without online search finished next in the average score ordering, above even the hand-set evaluation functions with online search. Among these three, the ordering was determine by the amount of training data. All of these evaluation functions were learned based on either self-play games or against a fixed set of opponents. The strong performance of these evaluation functions against the other strategies in this space is evidence that the learned strategies generalize to novel strategic contexts. This is consistent with the hypothesis of strategic independence in this set of parameters, and the ability of reinforcement learning to exploit this as a local search method.

The differences in the strategic analysis of these two strategy spaces also suggest that the attack angle is to some degree more strategically interesting than the other evaluation and search parameters. First, the evaluation/search space has pure-strategy equilibria, while the attack angle space does not. There is a much greater diversity of viable strategies in the 10-strategy attack angle space than the 17-strategy evaluation/search space, in which only 2 strategies factor into the equilibrium analysis. The summary statistics also reflect a more balanced strategy space for the attack angle space, while the evaluations/search space shows large discrepancies among the strategies on these aggregate measures. All of these observations are consistent with the objective of separating the strategy space into strategic and independent dimensions of the parameter space.

5.2 Candidate Exploration Policies

I now move to a more abstract setting, first describing several candidate policies for exploring games in the reveal-payoff setting. These policies build on the intuitions developed in the chaturanga case study. A baseline policy selects profiles to evaluate at random. The three remaining methods are based on the idea of finding a best-response to an *active* strategy profile. They differ in how this profile is chosen. I also test some variations of these algorithms that use approximate best response in place of exact best-response. The first two best-response methods have been studied in the literature, but are limited to confirming only pure-strategy Nash equilibria. I introduce a novel exploration method that can also confirm mixed-strategy equilibria.

5.2.1 Random Selection

The *Random Selection* (RS) policy serves as a baseline. At each iteration, this algorithm selects a random profile to evaluate from the set of unevaluated pure-strategy profiles. It tracks information about the stability of each pure-strategy profile, including the number of deviations evaluated and the best deviation observed so far. This entails looping over the set of unilateral deviations $\mathscr{D}(s)$ and updating statistics each time a new profile is evaluated.

The same data structure is used in all subsequent algorithms.

5.2.2 Tabu Best-Response Dynamics

Best-response dynamics has a long history as a method for finding equilibria of games (Fudenberg and Levine, 1998), going back at least to Cournot (1838). *Tabu best-response dynamics* (TBRD) was proposed by Sureka and Wurman (2005) for the problem of finding sample pure-strategy Nash equilibrium without searching the entire profile space. The algorithm starts by selecting a random pure-strategy profile as the *active* profile. It loops through the players, computing a best response for the player to the active profile. If an improving deviation is found, a new active profile is formed by replacing the player's strategy in the old profile with the best response. A Nash equilibrium is confirmed when the algorithm loops through all players without finding an improving deviation. A potential problem with this procedure is that it may become trapped in an infinite cycle among a set of profiles. Sureka and Wurman (2005) prevent this behavior using a *tabu list* to exclude any profile from becoming active more than once. Profiles are added to the list when they become active, and any deviation that results in a profile in this list is excluded as a possible best response.[11]

I base my implementation on the version of TBRD studied by Jordan et al. (2008), with minor modifications. For some of the tasks I consider, there is value in continuing exploration after an equilibrium is confirmed. I modify the algorithm to restart from a random profile in any case where there are no best responses that are not tabu (for example, after confirming an equilibrium). I also track the stability properties of all pure-strategy profiles using the same method as the RS policy. Pseudocode for my implementation of TBRD is given in Algorithm 1.

In the base version of the algorithm, the $SELECT\text{-}DEVIATION(s)$ subroutine evaluates

[11] Sureka and Wurman (2005) consider both an explicit memory version of the tabu list that remembers profiles, and an attribute-based version that remembers strategies; I consider only the explicit version of the algorithm in this work.

Algorithm 1 Tabu Best-Response Dynamics

```
$s \leftarrow$ random profile
$k \leftarrow$ last player
$\bar{\varepsilon} \leftarrow \infty$
**while** observation bound not reached **do**
    $i \leftarrow$ next player
    $s'_i \leftarrow SELECT\text{-}DEVIATION(s)$ from $\mathscr{D}_i(s) \backslash TABU$
    **if** improving deviation **then**
        $k \leftarrow i$
        add $s$ to $TABU$
        $s \leftarrow [s'_i, s_{-i}]$
    **else**
        **if** $i = k$ **then**
            $\bar{\varepsilon} \leftarrow \varepsilon(s)$
            $s \leftarrow$ random profile
        **end if**
    **end if**
**end while**
```

all deviations for player i to strategy profile s and selects the best one. I also experiment with variants of the algorithm that compute *improving* responses, rather than best responses. To compute an improving response, the $SELECT\text{-}DEVIATION(s)$ subroutine may not need to evaluate all of the deviations. Two parameters control this behavior. The *aspiration level* is the minimum benefit to deviating that must be found by $SELECT\text{-}DEVIATION(s)$ before returning an improving response. The *minimum deviations tested* parameter is a lower bound on the number of deviations that must be evaluated in each iteration. If either of these parameters is set to ∞, then $SELECT\text{-}DEVIATION(s)$ always returns an exact best response.

The order in which deviations are tested may affect the performance of the algorithm when using improving responses. In particular, it is desirable to test deviations that are more likely to be beneficial early to avoid unnecessary evaluations. In some case this is impossible, since the current data set need not contain any information about the payoffs for unevaluated profiles.[12] However, some games may have structure which allows generalization of payoff

[12] For example, if payoffs for each profile are drawn independently from some distribution, there is no way to predict which deviations are more likely to be beneficial.

information from evaluated profiles to predict the payoffs for unevaluated profiles.

The average payoff heuristic is one simple method that may be able to exploit such structure. Rather than evaluate potential deviations in random order, the heuristic uses the average payoffs for each strategy, taken over the evaluated profiles. Deviations are ordered by the average payoff for the strategy the deviating player changes *to*. The *evaluation density* parameter allows this ordering heuristic to be used only after enough profiles have been evaluated, specified as the total number of profiles per pure strategy.[13] Until the requisite number of have been evaluated, deviations are tested in random order; after the threshold is reached, they are ordered by average payoff. If the evaluation density is ∞, the algorithm always uses the random ordering.

5.2.3 Minimum-Regret-First Search

Minimum-Regret-First Search (MRFS) has been used in several prior applications of empirical game-theory (Vorobeychik et al., 2006; Kiekintveld et al., 2006b).[14] Jordan et al. (2008) conducted the first rigorous test of this algorithm on an equilibrium confirmation task, comparing against a version of tabu best-response. The central idea of MRFS is to maintain an hypothesis that a pure-strategy profile is a Nash equilibrium, and evaluate profiles which will confirm or reject this hypothesis as quickly as possible. The algorithm maintains a list of equilibrium *candidates*—profiles for which the payoffs are known, but which have untested deviations. It tracks the lower bound on regret for each candidate profile, denoted $\hat{\varepsilon}(s)$, using the same method as RS. At each iteration, the search selects an active profile s from the set of candidates with minimum $\hat{\varepsilon}$, and selects the next profile to evaluate from the unevaluated deviations of s. When all deviations for a profile have been evaluated, the value of ε is known and the profile is moved from the set of candidates to the set of *confirmed* profiles. As soon as a profile has all deviations evaluated with $\varepsilon \leq 0$, an equilibrium has

[13] Strategies may have varying representation in the evaluated profiles The threshold is on the total number of profiles, so there is no lower bound on the number of profiles evaluated containing any specific strategy.

[14] The method was called "Best-First Search" in these papers.

been found.

One area where the performance of MRFS can potentially be improved is the order in which deviations are tested. I test an ordering using the same average payoff heuristic introduced for TBRD. The heuristic orders the deviations using average payoffs after the threshold *evaluation density* is reached. A setting of ∞ corresponds to the case where the algorithm always selects deviations randomly. My implementation (Algorithm 2) is close to that of Jordan et al. (2008), with the addition of the average payoff heuristic for the $SELECT\text{-}DEVIATION(s)$ subroutine.

Algorithm 2 Minimum-Regret-First Search

```
Add random profile to candidate set
while observation bound not reached do
    Select candidate profile $s$ with minimum $\hat{\varepsilon}$
    if all deviations of $s$ are tested then
        move $s$ from candidates to confirmed
    else
        $s' \leftarrow SELECT\text{-}DEVIATION(s)$ from $\mathscr{D}_i(s)$
        if $s'_i$ is not explored then
            Insert $s'_i$ into candidates
            update $\hat{\varepsilon}$ for all $\bar{s} \in s' \cup \mathscr{D}(s')$ in candidate set
        end if
    end if
end while
```

5.2.4 Subgame Best-Response Dynamics

A significant limitation of both TBRD and MRFS is that they confirm only pure-strategy equilibria. This is particularly problematic for games where no pure-strategy equilibrium exists, since both algorithms will evaluate the entire profile space if allowed to do so. I introduce a novel algorithm that generalizes best-response dynamics to confirm mixed-strategy Nash equilibria in addition to pure-strategy equilibria. *Subgame Best-Response Dynamics* (SBRD) tracks an active subgame, rather than an active pure-strategy profile. A subgame is defined by a subset of each player's pure strategy set. An active equilibrium candidate (in pure or mixed strategies) is identified by solving for an equilibrium of the subgame, using a standard equilibrium solver. The search tests deviations from this equilibrium candidate to find a best response, evaluating profiles outside of the active subgame as necessary. If an improving response is found, the strategy is added to the subgame and the procedure iterates; if not, an equilibrium has been confirmed.

The most important feature of SBRD is the capability to confirm mixed-strategy Nash equilibria without evaluating the full game. In fact, the search procedure is guaranteed to confirm a Nash equilibrium if the size of the subgame is unrestricted. On each iteration of the algorithm, there are two possible outcomes:

1. A best-response is found, and the strategy is added to the subgame.
2. The current candidate is confirmed as a Nash equilibrium.

Each time a best response is found, the size of the subgame increases until it contains all strategies in the full game. At this point, the algorithm degenerates to running the equilibrium solver on the original game. There is no need to use a tabu list to prevent cycling in SBRD. In cases where best-response dynamics cycles between profiles, the strategies are simply added to the subgame and become part of the equilibrium computation.

When the active subgames for SBRD are small relative to the size of the full game, the equilibrium and best-response computations are relatively inexpensive in terms of the number of profiles evaluated.[15] Let l be the maximum size of any players' subgame strategy set, m be the maximum size of any players' full strategy set, and n be the number of players. The entire subgame must be evaluated to find an equilibrium using a standard solver; the size of the subgame is bounded by l^n. Since each iteration expands the previous subgame, some profiles are previously evaluated. Finding a best response to the current equilibrium candidate $\hat{\sigma}$ requires evaluating all profiles with positive probability in the mixture $[s_i', \hat{\sigma}_{-i}]$, for all pure strategies s_i' that are not in the subgame. The number of such profiles is bounded from above by $m \cdot l^{n-1}$, since $\hat{\sigma}_{-i}$ covers at most l^{n-1} profiles and there are m possible pure strategies to test. If the support of $\hat{\sigma}_{-i}$ does not include all strategies in the subgame, the

[15] Equilibrium computations are also fast for small subgames, but this is a secondary concern for the purposes of my evaluation.

number of profiles evaluated during the best-response operation may be much lower.

Fortunately, there are theoretical reasons to believe that equilibria can often be identified using small subgames. Work by McLennan and Berg (2005) on two-player games with payoffs drawn from specific distributions shows that small sets of pure strategies are more likely to form the support of a Nash equilibrium than larger sets. Approximate Nash equilibria with small (logarithmic) support are also known to exist in general settings (Lipton et al., 2003). Porter et al. (2008) exploit these results to develop a search procedure for computing sample Nash equilibria. Their primary interest is minimizing running time to compute equilibria, rather than the number of evaluated profiles, but the success of their approach provides further motivation for SBRD.

Pseudocode for SBRD is given in Algorithm 3. The *FIND-EQUILIBRIUM* subroutine can be implemented using any standard Nash equilibrium solver. I first use a simple enumeration to check for pure-strategy equilibrium candidates. In addition to being fast, pure-strategy candidates are desirable in that they require fewer evaluations to test for best responses. If no pure-strategy equilibrium exists, I fall back on the logit equilibrium solver implemented in Gambit (Turocy, 2005; McKelvey et al., 2006).[16] The *SELECT-DEVIATION*(σ) subroutine evaluates all possible deviations for a player and selects the best response. I use only exact computations in the existing implementation, but this could be extended to use improving responses like those implemented for TBRD.

It may be desirable to restrict the growth of the subgame, due to the increasing number of evaluations required for equilibrium and best-response computations as the size of the subgame increases. If the size of the subgame is permanently restricted to less than the size of the full game, the search is incomplete and is no longer guaranteed to find an equilibrium. However, the completeness holds as long as the subgame is eventually allowed to contain the full game. I implement restrictions on the size of the subgame as follows. At each iteration, there is a maximum number of strategies allowed in the subgame for each player.

[16] This solver does not guarantee finding an exact equilibrium, but in practice I have found this to the be most reliable of the available algorithms for computing sample mixed-strategy equilibria.

This is defined by two parameters: the *initial size* is the maximum size at the start, and the *growth rate* gives the number of best-response iterations for each player before the size is incremented by one. If the bound is violated after adding a new strategy to the subgame, the *PRUNE-SUBGAME* subroutine removes a strategy. My implementation removes the oldest strategy for the given player. A growth rate of less than one or an infinite initial size both imply no restrictions on the subgame. I explore variations of the growth rate experimentally.

Algorithm 3 Subgame Best-Response Dynamics

```
subgame ← random pure-strategy profile
$\bar{\varepsilon} \leftarrow \infty$
$\sigma \leftarrow$ subgame
$k \leftarrow$ last player
equilibriumFound ← **false**
**while** observation bound not reached AND NOT equilibriumFound **do**
  $i \leftarrow$ next player
  $s'_i \leftarrow SELECT\text{-}DEVIATION(\sigma)$
  **if** improving deviation **then**
    $k \leftarrow i$
    add $s'_i$ to subgame
    $PRUNE\text{-}SUBGAME$
    $\sigma \leftarrow FIND\text{-}EQUILIBRIUM$ (subgame)
  **else**
    **if** $i = k$ **then**
      $\bar{\varepsilon} \leftarrow \varepsilon(\sigma)$
      equilibriumFound ← **true**
    **end if**
  **end if**
**end while**
**loop**
  evaluate random profile
**end loop**
```

5.3 Game Classes

I experiment with several classes of games that have known structural properties. All of these classes have been defined elsewhere in the literature, and all but one are generated using the GAMUT toolkit (Nudelman et al., 2004). *Uniform random* and *constant sum* classes are identical to those defined in Section 4.2. Uniform random games are instances of GAMUT's random game class, and constant sum games are instances of covariance games with a covariance parameter of -1. These are baselines that should be challenging for exploration policies. There is no general compact representation for these games, and the payoffs across different profiles are not correlated. In addition, instances of games drawn from these classes often do not have pure-strategy Nash equilibria.

The remaining classes of games all have known structure which exploration policies may be able to exploit. In *congestion games*, players choose subsets of *facilities* and receive the sum of the payoffs for the chosen facilities. The payoffs associated with each facility depend only on the number of players that choose the facility. This payoff structure can be represented using a potential function, implying that the games always have a pure-strategy equilibrium (Rosenthal, 1973). Best-response dynamics is known to converge to an equilibrium in games with potential functions (Monderer and Shapley, 1996).

Random local-effect games (random LEGs) have local interactions among pure strategies (Leyton-Brown and Tennenholtz, 2003). These interactions are specified using a graphical structure, with each node representing a pure strategy and links between nodes representing an effect between strategies. A pure strategy s^1 affects strategy s^2 if the utility for players selecting s^2 depends on an arbitrary function of the number of players selecting s^1. I generate instances of LEGs using GAMUT's default parameter settings and random graphs. LEGs often (but do not always) have pure-strategy equilibria, and experiments show that best-response dynamics quickly converges to equilibrium for most instances of LEGs.

Supermodular games (Topkis, 1979) have received considerable attention in the literature, in part because many classic games belong to this class. Supermodularity is a complementarity condition on payoffs. A game is supermodular if the strategy sets for the players are partially ordered, and the marginal payoffs for selecting a higher strategy are increasing as other players select higher strategies. Milgrom and Roberts (1990) show that supermodular games possess pure-strategy Nash equilibria, and that a class of adaptive methods including best-response dynamics converges to the set of pure-strategy equilibria in the limit. The GAMUT class of supermodular games comprises parameterized versions of three classic games: arms race, Cournot duopoly, and Bertrand oligopoly. I use GAMUT's default randomization over instances of these classic games (including their parameter settings).

The final class of games I consider are *factored games* (Davis et al., 2007). A *product* game is composed from a set of *factor* games, all in normal form. Instances of the factored game class are product games. The strategy sets for a product game are defined by the cross product of the strategy sets in the factor games. Each pure strategy in the product game can be represented by a vector of strategy choices for each factor game. The payoffs for the product game are defined by adding the payoffs for the corresponding outcomes in the factor games.

The current version of GAMUT does not have a generator for factored games, so I generate instances as follows. Each product game has exactly two factors, each of which may have different numbers of pure strategies. One is the strategic factor, and may have arbitrary payoffs. The strategic factors I use are all instances of the random game class. The other is the independent factor, which represents a degenerate game in which players receive the same payoff for a given pure strategy, regardless of the other players' strategies. The optimal choices for independent factors can be determined in isolation. I generate independent factors by drawing payoffs for each pure strategy from the same distribution as the payoffs in the random game class. The instances of the factored game class are products of strategic and independent factors.

These games are an interesting case for exploration policies because they have an intermediate degree of structure. They have a compact representation in terms of the individual factors. However, they do not necessarily possess pure-strategy equilibria, and best-response dynamics and other adaptive methods may fail to converge to equilibria. In this sense, they are significantly more challenging than congestion, supermodular, and local-effect games.

The degree of independence structure in factored games varies with the relative number of pure strategies in the independent and strategic factors. When the strategic factor has fewer than two strategies, all players in the product game have a dominant strategy. When the independent factor has fewer then two strategies, the product game is equivalent to a uniform random game. My inclusion of this form of factored games in the experimental analysis is motivated part by the type of strategic independence hypothesized for the chaturanga strategy space.

5.4 Identifying Equilibrium

I first present experimental results comparing the performance of the candidate exploration policies on the *equilibrium confirmation* task for the revealed-payoff observation model. The objective in this task is to confirm a Nash equilibrium of the underlying game, evaluating as few profiles as possible. This is the central task studied by both Sureka and Wurman (2005) and Jordan et al. (2008). I also study the strategy selection task in the next section, but there are independent motivations for equilibrium confirmation. For instance, a central authority may wish to provide advice to all players in a game by suggesting an equilibrium that they could play.

I test the exploration policies on instances drawn from the six game classes described above. All games have two players, and are categorized as either medium or large depending on the number of pure strategies. Medium games have 25 pure strategies for each player (625 profiles), except congestion games which have 31 pure strategies (961 profiles).[17] Factored games of this size have five pure strategies for both the strategic and independent factors, yielding 25 strategies for the product game.

Large game instances have 100 pure strategies (10000 profiles). I test three variations of

[17] Since strategies for congestion games represent choices of subsets, the number of strategies is always $2^k - 1$ for some value of k where k represents the number of facilities. Five is the maximum number of facilities that GAMUT allows for congestion games.

Parameters	Random	Constant	Factored	LEG	Supermodular	Congestion
∞, ∞, ∞	389	625	247	76	135	**91**
0, 0, ∞	**381**	625	255	86	87	118
0.1, 0, ∞	383	625	246	77	87	105
0, 10, ∞	384	625	250	77	95	102
0.1, 10, ∞	384	625	246	76	95	97
0, 0, 0	382	625	247	82	82	113
0.1, 0, 0	386	625	**241**	**75**	**75**	96
0, 10, 5	378	625	253	88	88	116
0.1, 10, 5	382	625	246	76	76	98

Table 5.7 TBRD medium games. The average number of profiles evaluated by variations of TBRD before a Nash equilibrium is confirmed. The three parameters specify the aspiration level, minimum deviations tested, and evaluation density, respectively. The first line listed corresponds to the base TBRD algorithm.

large factored games with different sizes for the strategic and independent factors, effectively varying the degree of independence structure in the game. The number of pure strategies for each factor is listed in the results as an ordered pair $(\mathit{strategic}, \mathit{independent})$. My experimental data set comprises 500 instances for each class of medium games, and 100 for each class of large games. The exploration policies are run twice on each game instance, averaging the results. Results are presented in the following tables as the average number of profiles evaluated to confirm equilibrium. If no equilibrium is confirmed, the policy evaluates the entire profile space. The initial tables show results for different parameter settings of each exploration policy to show the impact of these settings. At the end I provide a summary table that compares the results for the different exploration policies.

The first two tables, 5.7 and 5.8, show results for the TBRD algorithm with various parameter settings. I highlight two interesting features of the data. TBRD confirms an equilibrium using many fewer evaluations for the classes with known structure than for random games or constant-sum games. The other striking feature is that varying the method of selecting deviations has very little impact on the performance of the policy in most cases. One exception is for supermodular games, where improving responses have better performance then best responses. Versions of TBRD with lower aspiration levels for improving

Parameters	Random	Factored (25,4)	Factored (10,10)	Factored (4,25)	LEG	Supermodular
∞, ∞, ∞	6729	4344	3457	2184	307	1480
0, 0, ∞	6491	4760	3583	2184	410	361
0.1, 0, ∞	6568	4459	3419	2132	325	499
0, 10, ∞	6657	4438	3415	2148	355	**348**
0.1, 10, ∞	6654	4220	**3410**	2112	312	488
0, 0, 0	**6419**	4218	3425	2197	379	373
0.1, 0, 0	6725	4248	3427	2112	**302**	513
0, 10, 5	6507	4291	3503	2172	373	362
0.1, 10, 5	6882	**4183**	3416	**2111**	314	492

Table 5.8 TBRD large games. The average number of profiles evaluated by variations of TBRD before a Nash equilibrium is confirmed. The three parameters specify the aspiration level, minimum deviations tested, and evaluation density, respectively. The first line listed corresponds to the base TBRD algorithm.

Parameters	Random	Constant	Factored	LEG	Supermodular	Congestion
∞	384	625	245	76	86	97
0	385	625	**241**	116	184	112
5	381	625	246	76	94	98
10	**375**	625	245	**75**	**85**	**96**

Table 5.9 MRFS medium games. The average number of profiles evaluated by variations of MRFS before a Nash equilibrium is confirmed. The parameter setting specifies the evaluation density for the average payoff heuristic. The first setting is the baseline random ordering.

responses consistently outperform versions with higher aspiration levels for this class of games. However, I observe the opposite effect with smaller magnitude for random LEGs, so this benefit depends on the type of structure present in the game. The claim with strongest support based on this evidence is that replacing best responses with improving responses does not substantially degrade the performance of TBRD.

For MRFS, the parameter space consists of variations of the average payoff heuristic. Results for variations of this parameter are shown in Tables 5.9 and 5.10. None of the versions of MRFS which use the average payoff ordering substantially outperform the baseline which uses random ordering. In some cases, the heuristic can harm performance when it is applied with very little data. This is most evident when the heuristic is applied with no threshold on the evaluation density (parameter setting 0), on games where equilibria

Parameters	Random	Factored (25,4)	Factored (10,10)	Factored (4,25)	LEG	Supermodular
∞	6968	4550	3564	2119	321	**527**
0	**6699**	**4373**	3584	**2078**	742	2564
5	6883	4473	3507	2139	313	1105
10	6725	4497	**3430**	2101	**310**	663

Table 5.10 MRFS large games. The average number of profiles evaluated by variations of MRFS before a Nash equilibrium is confirmed. The parameter setting specifies the evaluation density for the average payoff heuristic; the first setting is the baseline random ordering.

Parameters	Random	Constant	Factored	LEG	Supermodular	Congestion
1, 1	**263**	**548**	112	74	**131**	92
1, 2	278	574	113	74	132	**91**
1, 4	306	593	114	**73**	132	**91**
2, 2	271	571	**111**	74	133	92
2, 4	282	591	112	74	134	**91**

Table 5.11 SBRD medium games. The average number of profiles evaluated by variations of SBRD before a Nash equilibrium is confirmed. The first parameter is the initial size of the subgame, and the second is the number iterations before the size is incremented. The first line listed corresponds to the base SBRD algorithm.

are usually found quickly using the random ordering (LEG, supermodular, and congestion).

Tables 5.11 and 5.12 show data for various parameter settings of SBRD, corresponding to different restrictions on how the size of the subgame increases. Here too, the data show only marginal differences in performance for different parameter settings. The most noticeable effect is for random games, where faster growth (lower settings for the second parameter) results in somewhat better performance.

Parameters	Random	Factored (25,4)	Factored (10,10)	Factored (4,25)	LEG	Supermodular
1, 1	**3201**	**1220**	679	410	312	1512
1, 2	3955	1384	703	412	310	1467
1, 4	4604	1459	692	419	311	**1288**
2, 2	3772	1396	**669**	409	310	1473
2, 4	4690	1483	701	**404**	**307**	1533

Table 5.12 SBRD large games. The average number of profiles evaluated by variations of SBRD before a Nash equilibrium is confirmed. The first parameter is the initial size of the subgame, and the second is the number iterations before the size is incremented. The first line listed corresponds to the base SBRD algorithm.

Solver	Random	Constant	Factored	LEG	Supermodular	Congestion
RS	614	625	615	604	612	944
TBRD	378	625	241	75	**85**	**91**
MRFS	375	625	241	75	**85**	96
SBRD	**263**	**548**	**111**	**74**	131	**91**

Table 5.13 All exploration policies, medium games. The average number of profiles evaluated to confirm a Nash equilibrium for each of the algorithms, taking the minimum over all parameter settings for each class of games.

Solver	Random	Factored	Factored	Factored	LEG	Supermodular
		(25,4)	(10,10)	(4,25)		
RS	9961	9958	9958	9951	9919	9917
TBRD	6419	4183	3410	2111	**302**	**348**
MRFS	6699	4373	3430	2078	310	527
SBRD	**3201**	**1220**	**679**	**404**	307	1288

Table 5.14 All exploration policies, large games. The average number of profiles evaluated to confirm a Nash equilibrium for each of the algorithms, taking the minimum over all parameter settings for each class of games.

Tables 5.13 and 5.14 summarize the results of the experiments with all variations of the different exploration policies. This table also adds the RS algorithm as a baseline. The results listed for TBRD, MRFS, and SBRD are for the best parameter settings for each game class. In most cases the performance for the best setting is quite similar to the performance for the other settings, with some exceptions noted above.

All three of the exploration policies exhibit strong improvements over the baseline random selection policy. This is true to some degree even for random games, but the improvements for structured games are especially dramatic. In some of the large game classes, the exploration policies confirmed an equilibrium after evaluating less than 5% of the profile space (i.e., 500 profiles), on average. These improvements are not specific to a particular kind of structure, and are observed for all three policies across all of the varieties of structure tested.

Comparing the individual policies, the performance of TBRD and MRFS is virtually identical across the range of game classes tested. This confirms the findings of Jordan et al. (2008) comparing these two policies on equilibrium confirmation. Jordan et al. do find that MRFS outperforms TBRD on a variation of the task for approximate equilibrium; I do not test this variation here. For game classes that always (or nearly always) have pure-strategy equilibria, SBRD has similar performance to both TBRD and MRFS. The interesting difference is for classes where pure-strategy equilibria may not exist. SBRD shows strong performance gains in this case; this is especially evident in large factored games, where SBRD outperforms TBRD and MRFS by large margins. One other interesting case is supermodular games, where TBRD and MRFS outperform SBRD. The performance of SBRD on these games is on par with versions of TBRD that use exact best-response computations. It is likely that introducing improving responses into the SBRD algorithm could improve performance on this class of games, similar to the improvement observed for TBRD.

One of the more interesting findings from these experiments on the whole is that varying the details of the exploration policies has only a marginal impact on performance, with a few important exceptions highlighted in the previous discussion. Most of the parameterized variations of the algorithms had nearly identical performance to the originals, and even the difference in high-level search control between TBRD and MRFS did not have a great impact. There is a very large improvement over the baseline for using some form of best response to direct exploration, especially on structured games. The differences observed for variations of this idea are small in comparison. The most important exception to this is SBRD, which achieves substantial improvements in some game classes by confirming mixed-strategy equilibria in addition to pure-strategy equilibria.

5.5 Selecting Strategies

The results from the previous set of experiments demonstrate the value of the candidate exploration policies for the equilibrium confirmation task. Here I test the value of these policies for the strategy selection task, applying the meta-strategy analysis framework introduced in Chapter 3. In the strategy selection experiments presented in Chapter 4, the information players received to make strategy choices was generated by fixed policies. For the incomplete observation model, the policy is equivalent to the random selection baseline in this chapter. In this section I study meta-strategies that combine the exploration policies defined in this chapter with the best strategy selection method from Chapter 4, logit equilibrium (LES). The question is whether the directed exploration policies improve the strategy selections made by LES.

I test the default version of each candidate exploration policy in combination with LES. The exploration policy is run until reaching a given bound on the number of profiles evaluated. This observation is used by LES to make strategy selections, just as in the incomplete observation model. I use the same method to tune the parameter λ of the LES algorithm as before. For each combination of exploration policy, game class, and bound on profiles evaluations, I simulate a meta-game with 10 different settings of λ. The value of λ with the lowest homogeneous regret is used in the following experiments comparing the exploration policies.

I present results for five classes of games: uniform random, supermodular, random LEG, and two variations of factored games. All games have 2 players and 10 pure strategies per player (100 profiles). The two versions of factored games have strategic and independent factors of sizes (5,2) and (2,5). I vary the number of profiles the exploration policy is allowed to evaluate. All of the meta-games are estimated from 1,000 game instances, with exploration policies run once for each instance. The performance measure is homogeneous profile regret, as defined in Section 4.3.

The results of the meta-strategy analysis are shown in Figures 5.3, 5.4, 5.5, 5.6, and 5.7. All three directed exploration policies show improvements over the random selection policy. There are benefits to deviating from RS in almost all cases where exploration policies are allowed to evaluate more than 10 profiles, and in most cases the benefit to deviating is large. Among the three directed policies, there is no method that clearly outperforms the others. All three typically have very low homogeneous regret (less than 1% of the maximum payoff). The policy with lowest regret depends on the class of games, and the amount of exploration permitted. MRFS typically has the lowest regret on random games and the factored games with larger strategic factors. However, it has higher regret than TBRD and SBRD on supermodular games, and for small numbers of profiles evaluated in random LEG games and (2,5) factored games.

Given that TBRD and MRFS had very similar performance in the equilibrium confirmation task, it is not surprising that these policies perform similarly on strategy selection. However, SBRD does not significantly outperform TBRD and MRFS, even for classes of games where it was able to confirm equilibria using fewer profiles (random and factored). One possibility is that some of these exploration policies are gathering evidence that is useful for strategy selection, but not directly useful for confirming equilibrium. Evaluating the policies on related tasks such as finding approximate equilibria could provide additional explanations for the results observed in the strategy selection task. A noticeable trend in the data is that the homogeneous regret for SBRD tends to increase as more profile evaluations are allowed. This may be explained in part by the fact that SBRD defaults to random selection after it has confirmed an equilibrium, and random selection performs quite poorly. An alternate policy implementing random restarts or continued subgame expansion with new criteria might improve performance at this stage.

5.6 Discussion

The experiments presented here constitute the broadest evaluation to date of exploration policies for the reveal-payoff model of observation. They cover parameterizations of three policies, including the novel SBRD policy. The evaluation covers games with several different forms of structure, and tests two distinct tasks, equilibrium confirmation and strategy selection. The candidate exploration policies improve performance over the baseline in

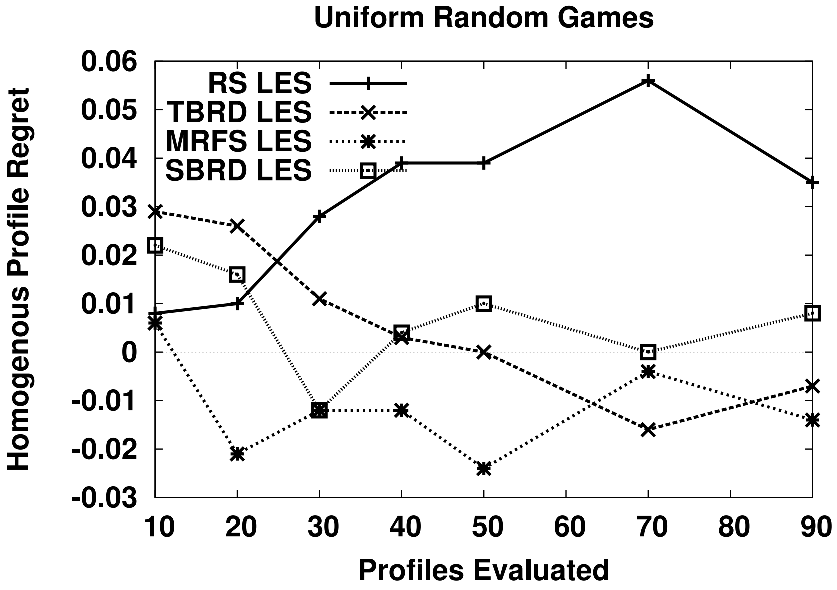


Figure 5.3 Meta-strategy analysis for candidate exploration policies on uniform random games.

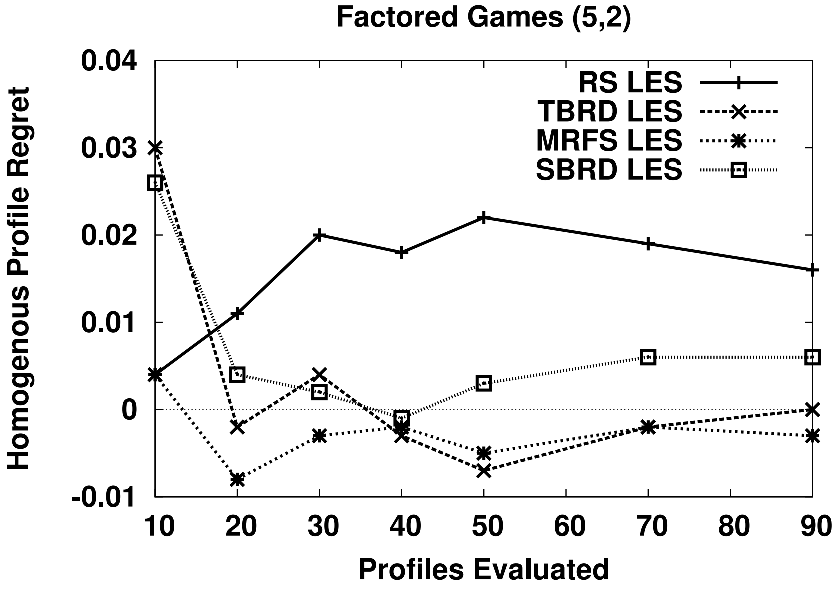


Figure 5.4 Meta-strategy analysis for candidate exploration policies on factored (5,2) games.

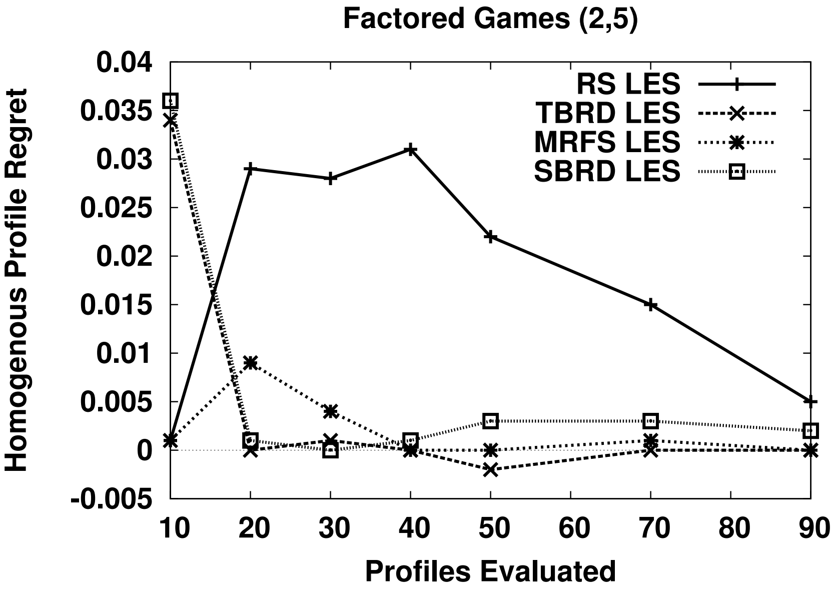


Figure 5.5 Meta-strategy analysis for candidate exploration policies on factored (2,5) games.

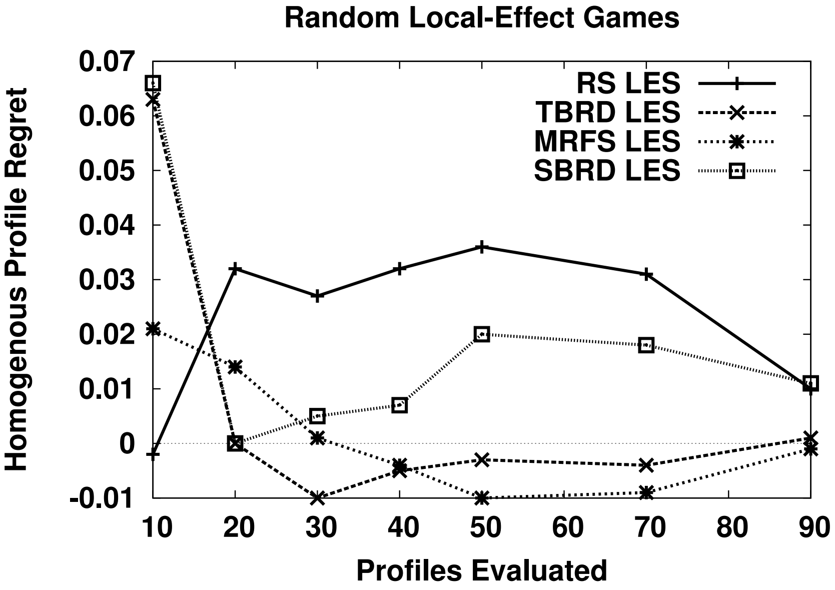


Figure 5.6 Meta-strategy analysis for candidate exploration policies on random LEG games.

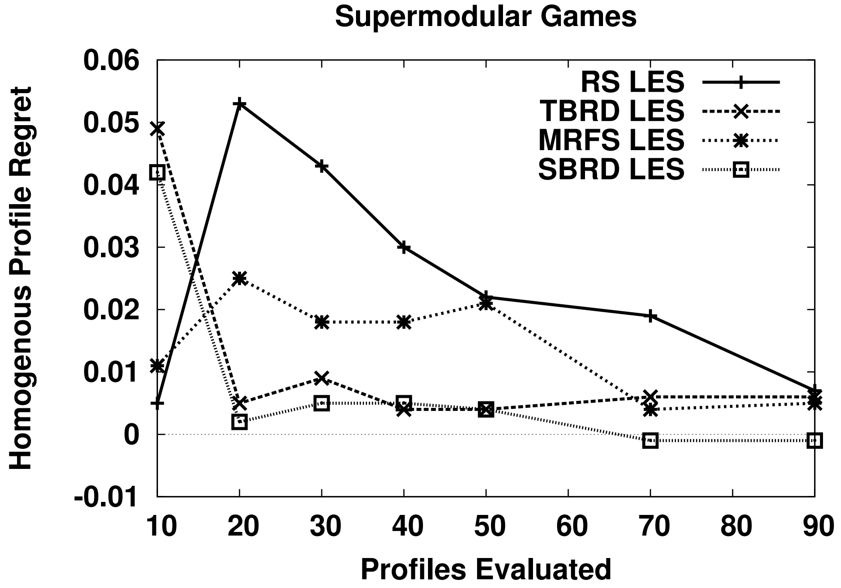


Figure 5.7 Meta-strategy analysis for candidate exploration policies on supermodular games.

on both tasks, and across all of the varieties of structure tested. In general, I find that the differences in performance between different variations of these policies is quite small, compared to the large gains from using any form of best response over the baseline random selection. The most interesting exception is for the confirmation task, where SBRD does offer clear improvements for some classes of games. The finding that most variations of the exploration policies are roughly equivalent may actually be quite useful in practice. Most straightforward policies that employ some form of best response are likely to improve performance, and the specific form of structure in the game does not need to be known or specified by the analyst up front to achieve these benefits.

The specific exploration policies and game classes explored in this chapter were motivated in part by the initial case study of chaturanga. It is possible to analyze chaturanga using an approach similar to SBRD. For a set of candidate chaturanga strategies, simulation can be used to estimate a game matrix for the subgame represented by these strategies.

Applying equilibrium analysis to this subgame identifies a set of opponents, and reinforcement learning can be used to search for an improving response to these opponents. The identified response is added to the set of candidates, and the process iterates. This process has the same high-level structure as SBRD, but uses simulation to estimate the payoffs in the subgame, and local search rather than exact best-response computations. My results provide evidence in an abstract setting that this form of exploration is likely to be effective if the parameterized strategy space for chaturanga has independence structure.

5.7 Related Work

Game playing programs have a long history in artificial intelligence (Schaeffer and van den Herik, 2002), going back to seminal works by Shannon (1950) on chess and Samuel (1959) on checkers. In most of this work, game theory plays little role in the analysis. My case study of chaturanga considers the game from a game-theoretic perspective. One area of game playing the does make significant use of game-theoretic principles is recent developments for poker-playing agents. Advances in exploiting symmetries and abstraction techniques have allowed researchers to find equilibrium solutions for increasingly large versions of poker (Zinkevich et al., 2008; Gilpin et al., 2008). Versions of the game with many players remain very challenging for current methods, but some progress has been made for versions with more than two players using approximation methods (Ganzfried and Sandholm, 2008).

In this chapter I study exploration policies for the case where payoffs are either revealed or not. Exploration policies for cases where profiles may be sampled to reveal noisy information about the payoffs are another interesting line of research. Walsh et al. (2003) introduced the first algorithm for this case, using an approximation of the value of information. Jordan et al. (2008) improves on this algorithm using a different approximation. Vorobeychik and Wellman (2008) also considers a similar task for infinite games, introducing a globally convergent methods based on simulated annealing.

Chapter 6

Conclusion

This thesis contributes to the development of practical tools for strategic reasoning in very complex games. My results advance the methodology of empirical game-theoretic analysis, which combines massive computation, empirical methods, and game-theoretic principles to reason about games that are well beyond the limits of conventional game-theoretic analysis. The existing literature related to empirical game-theoretic analysis describes the general methodology and several successful applications to specific domains. However, there has been little comprehensive evaluation of different techniques for building and analyzing empirical game models. I contribute to the literature by developing and evaluating methods for empirical game-theoretic analysis, emphasizing broad evaluation to identify general principles.

I model empirical game estimation as a meta-game, and apply this framework to derive new insights into effective approaches for both building and analyzing empirical game models. The first specific problem I consider is selecting a strategy given an uncertain model of the game. For this task, I identify logit equilibrium as the current champion and present considerable evidence that this method makes good predictions of opponent play across a wide range of conditions. I also consider the problem of selecting samples to build an estimated game model given limited observation capabilities. Here I provide a comprehensive evaluation of several exploration policies for games on both the strategy selection task and an equilibrium identification task. All of the policies tested substantially outperform a random exploration baseline for both tasks on games with known structural properties. A novel exploration policy I introduce—subgame best-response dynamics—is able to confirm mixed-strategy equilibria in addition to pure-strategy equilibria, generalizing previous methods. This policy substantially improves performance on an equilibrium confirmation task for some classes of games.

6.1 Summary of Contributions

Chapter 2 describes the methodology of empirical game-theoretic analysis and establishes the relevance of this approach for strategic reasoning in very challenging domains. I report an application of the methodology to the Trading Agent Competition Supply Chain Management game. Solving the full version of this game in any exact sense is well beyond present capabilities for game-theoretic analysis. To my knowledge, no other team of agent developers has even attempted to use game-theoretic reasoning to analyze the full TAC SCM game—or any similar scenario. This game pushes the limits even for empirical game-theoretic analysis, possessing an extremely large strategy space, stochastic outcomes, and high computational demands for simulating game instances. Nevertheless, we are able to draw useful conclusions about agent strategies using this approach, supplementing tournament results. This application contributes broadly to the evidence supporting the usefulness of empirical methods for analyzing specific complex games, and is particularly interesting as an example of the scalability of the methodology.

The remainder of the thesis advances our understanding of how to perform empirical game-theoretic analysis well. Chapter 3 begins by developing a common framework for evaluation, which to this point has been largely absent from the literature. The central insight is that the process of empirical game-theoretic analysis can be modeled as a *meta-game*, endogenizing the players' estimates of the game as observations. I use game-theoretic analysis of meta-strategies to evaluate a variety of approaches for building and analyzing empirical estimates of games. The methodology I apply is experimental, and designed to support broad evaluation across many game instances, different types of games, and different modes of observations. This approach is particularly useful for studying the strengths and weaknesses of meta-strategies across different conditions, yielding more general insights into their performance. The experiment that concludes the chapter demonstrates meta-strategy analysis and some of the capabilities of the approach. Using a relatively simple setup, I present compelling evidence that applying complete-information solution concepts to estimated games results in poor performance under uncertainty.

Chapter 4 addresses the question of how players should analyze estimated games to select strategies. In this setting, players have complex payoff uncertainty that does not vanish in the limit. Most theoretical treatments of payoff uncertainty either examine cases with infinitesimal noise, or make restrictive assumptions on the uncertainty that do not hold for the general case of estimating a game. Is it possible to offer any guidance at all for how to play using an estimated game model? I introduce three families of candidate meta-strategies that interpolate—using a single parameter—between uninformed predictions and equilibrium predictions that ignore noise in the estimated game. I apply the meta-strategy analysis framework from Chapter 3 to analyze these candidates, yielding two main results. First, I show evidence for a systematic relationship between observation uncertainty and the interpolation parameters. As uncertainty increases, broader predictions are preferable. Second, I find strong support for *logit equilibrium* as the champion for this task among the proposed candidates, identifying this as an appropriate tool for analyzing estimated game models. Both of these main results hold for a range of different game classes and observation models.

Chapter 5 is concerned with building good models to estimate a game. In particular, if a player has control over the information revealed, what policy should the player use to explore the profile space? The chapter begins with a motivating case study of chaturanga, focusing particularly on the idea of exploiting independence structure in a parameterized strategy space for the game. Building on the intuitions of this study, I test exploration policies on classes of games with known structural properties. I provide the most comprehensive evaluation to date of exploration policies for the revealed-payoff observation model, testing performance on both an equilibrium identification task and the strategy selection task (using the meta-game framework). A novel policy I propose, *subgame best-response dynamics*, is able to confirm mixed-strategy Nash equilibrium without exhaustive exploration. This policy improves the state of the art for equilibrium-finding algorithms on classes of games that may not possess pure-strategy equilibria. All of the candidate policies are generally able to exploit a variety of structural properties to improve performance on both tasks, relative to a random exploration baseline policy. I also explore different parameterizations of the base exploration policies for the equilibrium identification task. With a few exceptions, varying the details of the search procedure using these parameterizations had little effect on the ability of the algorithms to identify equilibria quickly. In particular, using approximate best-response calculations in lieu of exact ones does not substantially degrade performance in the cases tested.

6.2 Future Work

In this thesis I take significant steps towards the goal of developing a comprehensive toolkit for strategic reasoning in complex domains. My approach is computational, and advances the broader agenda of applying the methods of computer science to scale game-theoretic analysis to large multi-agent systems. One of the key challenges to advancing this agenda is to find ways to generalize the results achieved through computational means. There are many impressive examples of computational methods applied to analyze specific games of ambitious scope, including checkers (Schaeffer et al., 2005), chess (Campbell et al., 2002), and the our own analysis of the Trading Agent Competition, parts of which featured in Chapter 2. However, the strategies generated for specific games seem to offer relatively little insight into how to play other games of interest. My work on meta-strategy analysis moves towards providing more general insights into effective strategic reasoning by investigating broader classes of models.

The methodology I develop for analyzing meta-games provides a framework for addressing many open questions, and I have only begun to investigate the possible approaches for building and analyzing empirical game models. The results of my initial explorations motivate further work in several directions. My experiments on strategy selection under uncertainty identified logit equilibrium as the current champion. In the discussion of this method I highlighted several ways in which this solution concepts simplifies aspects of the meta-game model. A next step in this line of work is to develop new variations of quantal-response equilibrium that remove these restrictions. If new meta-strategies generated using this approach prove effective for playing meta-games, it would be very interesting to test whether these improvements also lead to better explanations of experimental data with human subjects. A similar extension of my work in this area is to add additional models from the behavioral literature as candidate meta-strategies, such as noisy introspection Goeree and Holt (2004) and cognitive hierarchies Camerer et al. (2004). My experimental framework is also well-suited to testing heuristics and other hypothesized behaviors that may be difficult to analyze mathematically.

Another important step in developing general approaches for strategy selection under uncertainty is to find better ways to select parameter settings for the algorithms. Most existing methods rely on fitting these parameters based on preliminary data. In my experiments I used the results of the self-play experiments to set parameters for the comparison across meta-strategies. Behavioral experiments often use maximum likelihood estimation to set parameters based on empirical data. These methods are not ideal, and it would be a significant advance to develop more principled methods for parameter choices based on what is known about the game and opponents. A final direction that would be useful to explore is finding exact solutions to (simple) meta-games. Even if these solutions are for restricted meta-games and do not generalize easily, they would be valuable for benchmarking other candidate meta-strategies to better evaluate the quality of the approximate solutions they offer.

In Chapter 5 I test exploration policies for cases where limited information about payoffs is available. These policies offer improved performance over random exploration on two different analysis tasks in cases where the game has underlying structure that can be exploited. One advantage of the policies tested is that they do not require any prior knowledge of specific structural properties. However, it seems likely that methods that explicitly identify and exploit specific types of structure could further improve performance on some analysis tasks. An active topic of research in game theory is developing structured representations of games and algorithms that exploit these representations. Most of the current research assumes that the game structure is provided by the analyst. If the analyst does not know the structure of the problem, this becomes a problem of structure *discovery*, and any structure must be inferred from the data. This is a challenging problem, but there are methods from single-agent decision theory that may inform structure discovery in games. For example, there are methods for learning the structure of Bayesian networks from empirical data (Friedman and Koller, 2003). One possible application of structure discovery is more efficient exploration policies for games. Structure discovery may also be of independent interest as a means of providing analysts as more intuitive understanding of the important strategic relationships in a game.

Bibliography

Aghassi, Michele and Dimitris Bertsimas. 2006. Robust game theory, *Mathematical Programming*, 107(1), 231–273.

Anderson, Simon P., Jacob K. Goeree, and Charles A. Holt. 2001. Minimum-effort coordination games: Stochastic potential and logit equilibrium, *Games and Economic Behavior*, 34, 177–199.

Arifovic, Jasmina, Richard D. McKelvey, and Svetlana Pevnitskaya. 2006. An initial implementation of the Turing tournament to learning in repeated two-person games, *Games and Economic Behavior*, 51(1), 93–122.

Armantier, Olivier, Jean-Pierre Florens, and Jean-Francois Richard. 2007. Approximation of Bayesian Nash equilibrium, Tech. rep., University of Montreal.

Arunachalam, Raghu and Norman M. Sadeh. 2005. The supply chain trading agent competition, *Electronic Commerce Research and Applications*, 4, 63–81.

Asada, Minoru, Hiroaki Kitano, Itsuki Noda, and Manuela Veloso. 1999. RoboCup: Today and tomorrow—What we have learned, *Artificial Intelligence*, 110(2), 193–214.

Aumann, Robert J. 1987. Correlated equilibrium as an expression of Bayesian rationality, *Econometrica*, 55(1), 1–18.

Axelrod, Robert. 1984. *The Evolution of Cooperation*, New York: Basic Books.

Bajari, Patrick, C. Lanier Benkard, and Jonathan Levin. 2007. Estimating dynamic models of imperfect competition, *Econometrica*, 75(5), 1331–1370.

Banerjee, Bikramjit and Peter Stone. 2007. General game learning using knowledge transfer, in *Twentieth International Joint Conference on Artificial Intelligence*, 672–677.

Basu, Kaushik and Jörgen W. Weibull. 1991. Strategy subsets closed under rational behavior, *Economic Letters*, 36(2), 141–146.

Baxter, Jonathan, Andrew Tridgell, and Lex Weaver. 2000. Learning to play chess using temporal-differences, *Machine Learning*, 40(3), 243–263.

Bednar, Jenna and Scott Page. 2007. Can game(s) theory explain culture? The emergence of cultural behavior within multiple games, *Rationality and Society*, 19(1), 65–97.

Beja, Avraham. 1992. Imperfect equilibrium, *Games and Economic Behavior*, 4(1), 18–36.

Benisch, Michael, Alberto Sardinha, James Andrews, and Norman Sadeh. 2006. CMieux: Adaptive strategies for competitive supply chain trading, in *Eighth International Conference on Electronic Commerce*, 47–58.

Bernheim, B. Douglas. 1984. Rationalizable strategic behavior, *Econometrica*, 52(4), 1007–1028.

Billings, Darse, Aaron Davidson, Jonathan Schaeffer, and Duane Szafron. 2002. The challenge of poker, *Artificial Intelligence*, 134(1–2), 201–240.

Blume, Lawrence E. 2003. How noise matters, *Games and Economic Behavior*, 44, 251–271.

Borghetti, Brett, Eric Sodomka, Maria Gini, and John Collins. 2006. A market-pressure-based performance evaluator for TAC-SCM, in *Agent-Mediated Electronic Commerce: Automated Negotiation and Strategy Design for Electronic Markets*, Springer Berlin/Heidelberg, no. 4452 in Lecture Notes in Artificial Intelligence, 178–188.

Brafman, Ronen I. and Moshe Tennenholtz. 2004. Efficient learning equilibrium, *Artificial Intelligence*, 159(1–2), 27–47.

Bresnahan, Timothy F. and Peter C. Reiss. 1991. Empirical models of discrete games, *Jounal of Econometrics*, 48, 57–81.

Cachon, Gerard P. and Serguei Netessine. 2004. Game theory in supply chain analysis, in David Simchi-Levi, S. David Wu, and Zuo-Jun (Max) Shen, (Eds.) *Handbook of Quantitative Supply Chain Analysis: Modeling in the eBusiness Era*, Kluwer, 13–66.

Camerer, Colin F., Teck-Hua Ho, and Juin-Kuan Chong. 2004. A cognitive hierarchy model of games, *The Quarterly Journal of Economics*, 119(3), 861–898.

Campbell, Murray, A. Joseph Hoane Jr., and Feng-Hsiung Hsu. 2002. Deep Blue, *Artificial Intelligence*, 134(1), 57–83.

Capra, C. Monica, Jacob K. Goeree, Rosario Gomez, and Charles A. Holt. 1999. Anomalous behavior in a traveler's dilemma?, *American Economic Review*, 89(3), 678–690.

Carlsson, Hans and Eric van Damme. 1993. Global games and equilibrium selection, *Econometrica*, 61(5), 989–1018.

Chatterjee, Kalyan and Terry P. Harrison. 1988. The value of information in competitive bidding, *European Journal of Operational Research*, 36(3), 322–333.

Chen, Yan and Yuri Khoroshilov. 2003. Learning under limited information, *Games and Economic Behavior*, 44, 1–25.

Cliff, Dave. 2003. Explorations in evolutionary design of online auction market mechanisms, *Electronic Commerce Research and Applications*, 2(2), 162–175.

Collins, John, Raghu Arunachalam, Norman Sadeh, Joakim Eriksson, Niclas Finne, and Sverker Janson. 2006. The Supply Chain Management Game for the 2007 Trading Agent Competition, Tech. Rep. CMU-ISRI-07-100, Carnegie Mellon University.

Collins, John, Wolfgang Ketter, Maria Gini, and Amrudin Agovic. 2007. Software architecture of the MinneTAC supply-chain trading agent, Tech. Rep. 07-006, University of Minnesota, Department of Computer Science.

Conitzer, Vincent and Tuomas Sandholm. 2003. Complexity results about Nash equilibrium, in *Eighteenth International Joint Conference on Artificial Intelligence*, 765–771.

Cournot, Augustin. 1838. *Researches into the Mathematical Principles of the Theory of Wealth*, Macmillan (1897), translation by Nathanial Bacon.

Daskalakis, Costantinos, Paul W. Goldberg, and Christos H. Papadimitriou. 2006. The complexity of computing a Nash equilibrium, in *Thirty-Eighth ACM Symposium on Theory of Computing*, 71–78.

Davis, George B., Michael Benisch, Kathleen M. Carley, and Norman M. Sadeh. 2007. Factoring games to isolate strategic interactions, in *Sixth International Joint Conference on Autonomous Agents and Multiagent Systems*, 443–449.

Dekel, Eddie and Drew Fudenberg. 1990. Rational behavior with payoff uncertainty, *Journal of Economic Theory*, 52, 243–267.

Erev, Ido and Alvin E. Roth. 1998. Predicting how people play games: Reinforcement learning in experimental games with unique mixed strategy equilibria, *American Economic Review*, 88(4), 848–881.

Erev, Ido, Alvin E. Roth, Robert L. Slonim, and Greg Barron. 2002. Predictive value and usefulness of game theoretic models, *International Journal of Forecasting*, 18(3), 359–368.

Eriksson, Joakim, Niclas Finne, and Sverker Janson. 2006. Evolution of a supply chain management game for the Trading Agent Competition, *Artificial Intelligence Communications*, 9, 1–12.

Foster, Dean and Peyton Young. 1990. Stochastic evolutionary game dynamics, *Theoretical Population Biology*, 38, 219–232.

Frankel, David M., Stepen Morris, and Ady Pauzner. 2003. Equilibrium selection in global games with strategic complementarities, *Journal of Economic Theory*, 108, 1–44.

Friedman, Daniel. 1996. Equilibrium in evolutionary games: Some experimental results, *The Economic Journal*, 106, 1–25.

Friedman, Nir and Daphne Koller. 2003. Being Bayesian about network structure: A Bayesian approach to structure discovery in Bayesian networks, *Machine Learning*, 50(1–2), 95–125.

Fudenberg, Drew, David M. Kreps, and David K. Levine. 1988. On the robustness of equilibrium refinements, *Journal of Economic Theory*, 44(2), 354–380.

Fudenberg, Drew and David K. Levine. 1998. *The Theory of Learning in Games*, MIT Press.

Fudenberg, Drew and Jean Tirole. 1991. *Game Theory*, MIT Press.

Ganzfried, Sam and Tuomas Sandholm. 2008. Computing an approximate jam/fold equilibrium for 3-player no-limit Texas Hold'em tournaments, in *Seventh International Conference on Autonomous Agents and Multiagent Systems*, 919–928.

Genesereth, Michael, Nathanial Love, and Barney Pell. 2005. General game playing: Overview of the AAAI competition, *AI Magazine*, 26(2), 62–72.

Gilpin, Andrew, Tuomas Sandholm, and Troels Bjerre Sorensen. 2008. A heads-up no-limit Texas Hold'em poker player: Discretized betting models and automatically generated equilibrium-finding programs, in *Seventh International Conference on Autonomous Agents and Multiagent Systems*, 911–918.

Glickman, Mark. 1995. Chess rating systems, *American Chess Journal*, 3, 59–102.

Goeree, Jacob K. and Charles A. Holt. 2004. A model of noisy introspection, *Games and Economic Behavior*, 46, 365–382.

Goeree, Jocob K. and Charles A. Holt. 2001. Ten little treasures of game theory and ten intuitive contradictions, *American Economic Review*, 91(5), 1402–1422.

Haile, Philip A., Ali Hortacsu, and Grigory Kosenok. 2008. On the empirical content of quantal response equilibrium, *American Economic Review*, 91(1), 180–200.

Harsanyi, John. 1967–1968. Games with incomplete information played by Bayesian players, Parts I, II, and III, *Management Science*, 14, 159–182, 320–334, 486–502.

Harsanyi, John. 1973. Games with randomly disturbed payoffs: A new rationale for mixed-strategy equilibrium points, *International Journal of Game Theory*, 2, 1–23.

He, Minghua, Alex Rogers, Xudong Luo, and Nicholas R. Jennings. 2006. Designing a successful trading agent for supply chain management, in *Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 1159–1166.

Hopkins, Ed. 2002. Two competing models of how people learn in games, *Econometrica*, 70(6), 2141–2166.

Jiang, Albert and Kevin Leyton-Brown. 2006. A polynomial-time algorithm for action-graph games, in *Artificial Intelligence*, 679–684.

Jordan, Patrick R., Christopher Kiekintveld, Jason Miller, and Michael P. Wellman. 2006. Market efficiency, sales competition, and the bullwhip effect in the TAC SCM tournaments, in *AAMAS-06 Workshop on Trading Agent Design and Analysis (TADA/AMEC)*, 62–74.

Jordan, Patrick R., Christopher Kiekintveld, and Michael P. Wellman. 2007. Empirical game-theoretic analysis of the TAC supply chain game, in *Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 1188–1195.

Jordan, Patrick R., Yevgeniy Vorobeychik, and Michael P. Wellman. 2008. Searching for approximate equilibria in empirical games, in *Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 1063–1070.

Kaiser, David M. 2007. Automatic feature extraction for autonomous general game playing agents, in *Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 655–661.

Kandori, Michihiro, George J. Mailath, and Rafael Rob. 1993. Learning, mutation, and long run equilibria in games, *Econometrica*, 61(1), 2956.

Kearns, Michael, Michael Littman, and Satinder Singh. 2001. Graphical models for game theory, in *Seventeenth Conference on Uncertainty in Artificial Intelligence*, 253–260.

Ketter, Wolfgang. 2007. *Identification and Prediction of Economic Regimes to Guide Decision Making in Multi-Agent Marketplaces*, Ph.D. thesis, University of Minnesota.

Kiekintveld, Christopher, Jason Miller, Patrick Jordan, and Michael P. Wellman. 2006a. Controlling a supply chain agent using value-based decomposition, in *Seventh ACM Conference on Electronic Commerce*, 208–217.

Kiekintveld, Christopher, Jason Miller, Patrick R. Jordan, Lee F. Callender, and Michael P. Wellman. 2008. Forecasting market prices in a supply chain game, *Electronic Commerce Research and Applications*. To appear.

Kiekintveld, Christopher and Michael P. Wellman. 2008. Selecting strategies using empirical game models: An experimental analysis of meta-strategies, in *Seventh International Joint Conference on Autonomous Agents and Multi-Agent System*, 1095–1102.

Kiekintveld, Christopher, Michael P. Wellman, and Satinder Singh. 2006b. Empirical game-theoretic analysis of chaturanga, in *AAMAS-06 Workshop on Game Theoretic and Decision Theoretic Agents*, 18–25.

Kiekintveld, Christopher, Michael P. Wellman, Satinder Singh, Joshua Estelle, Yevgeniy Vorobeychik, Vishal Soni, and Matthew Rudary. 2004. Distributed feedback control for decision making on supply chains, in *Fourteenth International Conference on Automated Planning and Scheduling*, 244–252.

Kiekintveld, Christopher, Michael P. Wellman, and Yevgeniy Vorobeychik. 2006c. An analysis of the 2004 supply chain management Trading Agent Competition, in *Agent-Mediated Electronic Commerce: Designing Trading Agents and Mechanisms*, Springer-Verlag, no. 3937 in Lecture Notes in Artificial Intelligence, 99–112.

Koller, Daphne and Brian Milch. 2003. Multi-agent influence diagrams for representing and solving games, *Games and Economic Behavior*, 45(1), 181–221.

Kontogounis, Ioannis, Kyriakos C. Chatzidimitriou, Andreas L. Symeonidis, and Pericles A. Mitkas. 2006. A robust agent design for dynamic SCM environments, in *Agent-Mediated Electronic Commerce: Designing Trading Agents and Mechanisms*, Springer Berlin/Heidelberg, vol. 3955 of *Lecture Notes in Computer Science*, 127–136.

Kreps, David M. and Robert Wilson. 1982. Sequential equilibrium, *Econometrica*, 50(4), 863–894.

Leyton-Brown, Kevin and Moshe Tennenholtz. 2003. Local-effect games, in *Eighteenth International Joint Conference on Artificial Intelligence*, 772–780.

Lipson, Asher. 2005. *An Empirical Evaluation of Multiagent Learning Algorithms*, Master's thesis, University of British Columbia.

Lipton, Richard J., Evangelo Markakis, and Aranyak Mehta. 2003. Playing large games using simple strategies, in *Fourth ACM Conference on Electronic Commerce*, 36–41.

Lise, Wietze. 2001. Estimating a game theoretic model, *Computational Economics*, 18, 141–157.

Luckhardt, Carol A. and Keki B. Irani. 1986. An algorithm for the solution of N-person games, in *Fifth National Conference on Artificial Intelligence*, 158–162.

Lustrek, Mitja, Matjaz Gams, and Ivan Bratko. 2005. Why minimax works: An alternative explanation, in *Nineteenth International Joint Conference on Artificial Intelligence*, 212–217.

MacKie-Mason, Jeffrey K., Anna Osepayshvili, Daniel M. Reeves, and Michael P. Wellman. 2004. Price prediction strategies for market-based scheduling, in *Fourteenth International Conference on Automated Planning and Scheduling*, 244–252.

Mackie-Mason, Jeffrey K. and Michael P. Wellman. 2006. Automated markets and trading agents, in Leigh Tesfatsion and Kenneth L. Judd, (Eds.) *Handbook of Computational Economics, vol. 2: Agent-Based Computational Economics*, North-Holland.

Mailath, George. 1998. Do people play Nash equilibrium? Lessons from evolutionary game theory, *Journal of Economic Literature*, 36(3), 1347–1374.

McKelvey, Richard D., Andrew M. McLennan, and Theodore L. Turocy. 2006. Gambit: Software tools for game theory, version 0.2006.01.20. Http://econweb.tamu.edu/gambit.

McKelvey, Richard D. and Thomas R. Palfrey. 1995. Quantal response equilibria for normal form games, *Games and Economic Behavior*, 10, 6–38.

McLennan, Andrew and Johannes Berg. 2005. Asymptotic expected number of Nash equilibria of two-player normal form games, *Games and Economic Behavior*, 51, 264–295.

Meyer, Yoella Bereby and Alvin E. Roth. 2006. The speed of learning in noisy games: Partial reinforcement and the sustainability of cooperation, *American Economic Review*. Forthcoming.

Milgrom, Paul and John Roberts. 1990. Rationalizability, learning, and equilibrium in games with strategic complementarities, *Econometrica*, 58(6), 1255–1277.

Milgrom, Paul R. and Robert J. Weber. 1982. A theory of auctions and competitive bidding, *Econometrica*, 50(5), 1089–1122.

Monderer, Dov and Lloyd S. Shapley. 1996. Potential games, *Games and Economic Behavior*, 14, 124–143.

Morris, Stephen and Hyun Song Shin. 2003. *Global games: Theory and applications*, vol. Advances in Economics and Econometrics (Proceedings of the Eighth World Congress of the Econometric Society), Cambridge University Press.

Myerson, Roger B. 1978. Refinements of the Nash equilibrium concept, *International Journal of Game Theory*, 7, 73–80.

Nash, John F. 1951. Non-cooperative games, *Annals of Mathematics*, 54, 298–295.

Nudelman, Eugene, Jennifer Wortman, Kevin Leyton-Brown, and Yoav Shoham. 2004. Run the GAMUT: A comprehensive approach to evaluating game-theoretic algorithms, in *Third International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 880–887.

Oechssler, Jorg and Burkhard Schipper. 2003. Can you guess the game you are playing?, *Games and Economic Behavior*, 43, 137–152.

Osborne, Martin J. 2004. *An Introduction to Game Theory*, Oxford University Press.

Osepayshvili, Anna, Michael P. Wellman, Daniel M. Reeves, and Jeffrey K. MacKie-Mason. 2005. Self-confirming price prediction for bidding in simultaneous ascending auctions, in *Twenty-First Conference on Uncertainty in Artificial Intelligence*, 441–449.

Papadimitriou, Christos H. and Tim Roughgarden. 2005. Computing equilibria in multiplayer games, in *Sixteenth ACM-SIAM Symposium on Discrete Algorithms*, 82–91.

Pardoe, David and Peter Stone. 2006. TacTex-05: A champion supply chain management agent, in *Twenty-First National Conference on Artificial Intelligence*, 1489–94.

Pardoe, David and Peter Stone. 2007. Adapting price predictions in TAC SCM, in *AAMAS-07 Workshop on Agent Mediated Electronic Commerce*, 29–42.

Pardoe, David and Peter Stone. 2008. An autonomous agent for supply chain management, in *Handbooks in Information Systems Series: Business Computing*, Elsevier. To appear.

Park, Juyong and M. E. J. Newman. 2005. A network-based ranking system for American college football, *Journal of Statistical Mechanics*.

Pearce, David G. 1984. Rationalizable strategic behavior and the problem of perfection, *Econometrica*, 52, 1029–1050.

Pell, Barney. 1993. *Strategy Generation and Evaluation for Meta-Game Playing*, Ph.D. thesis, University of Cambridge.

Phelps, Steve, Marek Marcinkiewicz, Simon Parsons, and Peter McBurney. 2005. Using population-based search and evolutionary game theory to acquire better-response strategies for the double-auction market, in *IJCAI-05 Workshop on Trading Agent Design and Analysis*, 21–27.

Phelps, Steve, Marek Marcinkiewicz, Simon Parsons, and Peter McBurney. 2006. A novel method for automatic strategy acquisition in *n*-player non-zero-sum games, in *Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 705–712.

Porter, Ryan W., Eugene Nudelman, and Yoav Shoham. 2008. Simple search methods for finding a Nash equilibrium, *Games and Economic Behavior*, 63(2), 642–662.

Powers, Rob and Yoav Shoham. 2004. New criteria and a new learning algorithm for learning in multi-agent systems, in *Seventeenth Advances in Neural Information Processing Systems*, 1089–1096.

Reeves, Daniel M. 2005. *Generating Trading Agent Strategies: Analytic and Empirical Methods for Infinite and Large Games*, Ph.D. thesis, University of Michigan.

Reeves, Daniel M. and Michael P. Wellman. 2004. Computing best-response strategies in infinite games of incomplete information, in *Twentieth Conference on Uncertainty in Artificial Intelligence*, 470–478.

Reisinger, Joseph, Erkin Bahceci, Igor Karpov, and Risto Miikkulainen. 2007. Coevolving strategies for general game playing, in *IEEE Symposium on Computational Intelligence and Games*, 320–327.

Rosenthal, Robert W. 1973. A class of games possessing pure-strategy Nash equilibria, *International Journal of Game Theory*, 2(1), 65–67.

Ross, Sheldon M. 2002. *Simulation*, Academic Press, third edn.

Roth, Alvin E. 2002. The economist as engineer: Game theory, experimental economics and computation as tools of design economics, *Econometrica*, 70(4), 1341–1378.

Samuel, Arthur L. 1959. Some studies in machine learning using the game of checkers, *IBM Journal of Research and Development*, 3, 211–229.

Sarin, Rajiv and Farshid Vahid. 2001. Predicting how people play games: A simple dynamic model of choice, *Games and Economic Behavior*, 34, 104–122.

Schaeffer, Jonathan, Yngvi Bjornsson, Neil Burch, Akihiro Kishimoto, Martin Muller, Rob Lake, Paul Lu, and Steve Sutphen. 2005. Solving checkers, in *Ninteenth International Joint Conference on Artificial Intelligence*, 292–297.

Schaeffer, Jonathan and H. Jaap van den Herik. 2002. Games, computers, and artificial intelligence, *Artificial Intelligence*, 134, 1–7.

Schraudolph, Nicol N., Peter Dayan, and Terrence J. Sejnowski. 1994. Temporal difference learning of position evaluation in the game of go, in *Sixth Advances in Neural Information Processing Systems*, 817–824.

Selten, Reinhard. 1975. Reexemination of the perfectness concept for equilibrium points in extensive games, *International Journal of Game Theory*, 4, 25–55.

Shannon, Claude E. 1950. Programming a computer for playing chess, *Philosophical Magazine*, 41, 256–275.

Shoham, Yoav, Rob Powers, and Trond Grenager. 2007. If multi-agent learning is the answer, what is the question?, *Artificial Intelligence*, 171(7), 365–377.

Slade, Margaret E. 1995. Empirical games: The oligopoly case, *The Canadian Journal of Economics*, 28(2), 368–402.

Smith, John Maynard. 1982. *Evolution and the Theory of Games*, Cambridge, MA: Cambridge University Press.

Sodomka, Eric, John Collins, and Maria Gini. 2007. Efficient statistical methods for evaluating trading agent performance, in *Twenty-Second Conference on Artificial Intelligence*, 770–775.

Stan, Mihai, Bogdan Stan, and Adina M. Florea. 2006. A dynamic strategy agent for supply chain management, in *Symbolic and Numeric Algorithms for Scientific Computing*, 227–232.

Stefani, Raymond T. 1997. Survey of the major world sports rating systems, *Journal of Applied Statistics*, 24(6), 635–646.

Sturtevant, Nathan. 2003. *Multi-Player Games: Algorithms and Approaches*, Ph.D. thesis, University of California, Los Angeles.

Sureka, Ashish and Peter R. Wurman. 2005. Using tabu best-response search to find pure strategy Nash equilibria in normal form games, in *Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 1023–1029.

Sutton, Richard S. and Andrew G. Barto. 1998. *Reinforcement Learning: An Introduction*, MIT Press/Bradford Books.

Taylor, P. and L. Jonker. 1978. Evolutionary stable strategies and game dynamics, *Mathematical Biosciences*, 16, 76–83.

Tennenholtz, Moshe. 2002. Competitive safety analysis: Robust decision-making in multiagent systems, *Artificial Intelligence*, 17, 363–378.

Tesfatsion, Leigh and Kenneth L. Judd, (Eds.) . 2006. *Handbook of Agent-Based Computational Economics*, Elsevier.

Thrun, Sebastian. 1995. Learning to play the game of chess, in *Seventh Advances in Neural Information Processing Systems*, 1069–1076.

Topkis, Donald M. 1979. Equilibrium points in nonzero-sum N-person submodular games, *SIAM Journal of Control and Optimization*, 17, 773–787.

Tumer, Kagan and Adrian Agogino. 2007. Distributed agent-based air traffic flow management, in *Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 341–349.

Turocy, Theodore L. 2005. A dynamic homotopy interpretation of the logistic quantal response equilibrium correspondence, *Games and Economic Behavior*, 51, 243–263.

Ulrich, Jan. 2006. An analysis of the 2005 TAC SCM finals, Tech. rep., University of Texas at Austin. Undergraduate thesis.

von Neumann, John and Oskar Morgenstern. 1944. *Theory of Games and Economic Behavior*, Princeton University Press.

Vorobeychik, Yevgeniy, Christopher Kiekintveld, and Michael P. Wellman. 2006. Empirical mechanism design: Methods, with application to a supply-chain scenario, in *Seventh ACM Conference on Electronic Commerce*, 306–315.

Vorobeychik, Yevgeniy and Daniel M. Reeves. 2008. Equilibrium analysis of dynamic bidding in sponsored search auctions, *International Journal of Electronic Business*. To appear.

Vorobeychik, Yevgeniy, Daniel M. Reeves, and Michael P. Wellman. 2007a. Constrained automated mechanism design for infinite games of incomplete information, in *Twenty-Third Conference on Uncertainty in Artificial Intelligence*, 400–407.

Vorobeychik, Yevgeniy and Michael P. Wellman. 2006. Mechanism design based on beliefs about responsive play (position paper), in *ACM EC-06 Workshop on Alternative Solution Concepts for Mechanism Design*.

Vorobeychik, Yevgeniy and Michael P. Wellman. 2008. Stochastic search methods for Nash equilibrium approximation in simulation-based games, in *Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 1055–1062.

Vorobeychik, Yevgeniy, Michael P. Wellman, and Satinder Singh. 2007b. Learning payoff functions for infinite games, *Machine Learning*, 67(2), 145–168.

Walsh, William E., Rajarshi Das, Gerald Tesauro, and Jeffery O. Kephart. 2002. Analyzing complex strategic interactions in multi-agent systems, in *AAAI-02 Workshop on Game-Theoretic and Decision-Theoretic Agents*.

Walsh, William E., David Parkes, and Rajarshi Das. 2003. Choosing samples to compute heuristic-strategy Nash equilibrium, in *AAMAS-03 Workshop on Agent-Mediated Electronic Commerce*.

Weibull, Jorgen. 2004. Testing game theory, in Steffen Huck, (Ed.) *Advances in Understanding Strategic Behavior: Game Theory, Experiments and Bounded Rationality.*, Palgrave MacMillan, 85–104.

Wellman, Michael P., Joshua Estelle, Satinder Singh, Yevgeniy Vorobeychik, Christopher Kiekintveld, and Vishal Soni. 2005a. Strategic interactions in a supply chain game, *Computational Intelligence*, 21, 1–26.

Wellman, Michael P., Amy Greenwald, and Peter Stone. 2007. *Autonomous Bidding Agents: Strategies and Lessons from the Trading Agent Competition*, MIT Press.

Wellman, Michael P., Amy Greenwald, Peter Stone, and Peter R. Wurman. 2003. The 2001 Trading Agent Competition, *Electronic Markets*, 13(1), 4–12.

Wellman, Michael P., Daniel M. Reeves, Kevin M. Lochner, Shih-Fen Cheng, and Rahul Suri. 2005b. Approximate strategic reasoning through hierarchical reduction of large symmetric games, in *Twentieth National Conference on Artificial Intelligence*, 502–508.

Wellman, Michael P., Peter R. Wurman, Kevin O'Malley, Roshan Bangera, Shou-de Lin, Daniel M. Reeves, and William E. Walsh. 2001. Designing the market game for a trading agent competition, *IEEE Internet Computing*, 5(2), 43–51.

Wray, Robert E., John E. Laird, Andrew Nuxoll, Devvan Stokes, and Alex Kerfoot. 2005. Synthetic adversaries for urban combat training, *AI Magazine*, 26(3), 82–92.

Zhang, Dongmo, Kanghua Zhao, Chia-Ming Liang, Gonelur Begum Huq, and Tze-Haw Huang. 2004. Strategic trading agents via market modelling, *SIGecom Exhanges*, 4(3), 46–55.

Zinkevich, Martin, Michael Johanson, Michael Bowling, and Carmelo Piccione. 2008. Regret minimization in games with incomplete information, in *Twentieth Advances in Neural Information Processing Systems*. To appear.