Generating Trading Agent Strategies: Analytic and Empirical Methods for Infinite and Large Games

by

Daniel M. Reeves

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
2005

Doctoral Committee:
Professor Michael P. Wellman, Chair
Professor Jeffrey K. MacKie-Mason
Professor Scott E. Page
Associate Professor Satinder Singh Baveja

First edition

Artificial Intelligence Laboratory
University of Michigan
1101 Beal Avenue
Ann Arbor, Michigan, 48109
USA

dreeves@umich.edu

http://ai.eecs.umich.edu/people/dreeves

To my parents, Laurie Kalkman Reeves and Martin Reeves.

Abstract

A *Strategy Generation Engine* is a system that reads a description of a game or market mechanism and outputs strategies for participants. Ideally, this means a game solver—an algorithm to compute Nash equilibria. This is a well-studied problem and very general solutions exist, but they can only be applied to small, finite games. This thesis presents methods for finding or approximating Nash equilibria for infinite games, and for intractably large finite games.

First, I define a broad class of one-shot, two-player infinite games of incomplete information. I present an algorithm for computing best-response strategies in this class and show that for many particular games the algorithm can be iterated to compute Nash equilibria. Many results from the game theory literature are reproduced—automatically—using this method, as well as novel results for new games.

Next, I address the problem of finding strategies in games that, even if finite, are larger than what any exact solution method can address. Our solution involves (1) generating a small set of candidate strategies, (2) constructing via simulation an approximate payoff matrix for the simplification of the game restricted to the candidate strategies, (3) analyzing the empirical game, and (4) assessing the quality of solutions with respect to the underlying full game. I leave methods for generating candidate strategies domain-specific and focus on methods for taming the computational cost of empirical game generation. I do this by employing Monte Carlo variance reduction techniques and introducing a technique for approximating many-player games by reducing the number of players. We are additionally able to solve much larger payoff matrices than the current state-of-the-art solver by exploiting symmetry in games.

I test these methods in small games with known solutions and then apply them to two realistic market scenarios: Simultaneous Ascending Auctions (SAA) and the Trading Agent Competition (TAC) travel-shopping game. For these domains I focus on two key price prediction approaches for generating candidate strategies for empirical analysis: self-confirming price predictions and Walrasian equilibrium prices. We find that these are highly effective strategies in SAA and TAC, respectively.

Acknowledgments

This thesis includes joint work with Michael Wellman, Jeffrey MacKie-Mason, Anna Osepayshvili, Kevin Lochner, Shih-Fen Cheng, Rahul Suri, Yevgeniy Vorobeychik, and Maxim Rytin. In Section 1.4 I outline this thesis in relation to previously published work with the above coauthors. My collaboration with Maxim Rytin, though, is published here for the first time as the proofs of key theorems in Chapter 3. Kevin O'Malley was instrumental in the development of our entry in the Trading Agent Competition, which is the subject of Chapter 6. Finally, William Walsh collaborated on the proofs of theorems related to the supply-chain game in Chapter 2.

I'm grateful to my thesis committee—Michael Wellman, Jeffrey MacKie-Mason, Scott Page, and Satinder Singh Baveja—for their careful reading, insightful comments, and cutting questions. This thesis is definitively improved due to their input. As early as my proposal defense, Satinder helped shape the course of this research by proposing what led to our epsilon metric, defined in Chapter 3. Similarly, Scott has proposed myriad improvements and helped clarify the contribution in Chapter 2 by asking hard questions about the generality of my approach. Even before Scott came to Michigan, he provided invaluable guidance at the Computational Economics Workshop at the Santa Fe Institute in 2000, when I first got excited about strategy generation in auctions. Jeff, of course, is much more than a committee member. As a coauthor of most of the material in Chapters 3, 4, and 5, it's thanks in no small part to his vision and hard work that this thesis is where it is.

This brings me to Mike. Perhaps it's filial bias but I am convinced there is no better research advisor in all of academia. Mike's research and thesis advising is nothing short of a graduate student's dream. He is an academic role model in every way, and I just can't thank him enough for his guidance and patience. Guidance doesn't capture it—he was a full collaborator (including writing code, running experiments, and proving theorems) on everything in this thesis. Patience is likewise an understatement; if he was ever frustrated by my almost explicit goal of stretching my graduate school career out as long as possible, he didn't let on. I also won't forget how kind he was when I was ill in 2002.

Others who I would like to thank for their help, support, or guidance include Terence Kelly, Andrzej Kozlowski, Daniel Lichtblau, Ted Turocy, Benjamin Grosof, David Reiley, Leslie Fine, Kayyut Chen, Jaap Suermondt, Evan Kirshenbaum, Kemal Guler, John Miller, Annie Noone, Bethany Soule, Dave Morris, Chris Kapusky, Karen Conneely, Sarah Nuss-Warren, Rachel Rose, Rob Felty, Michelle Sternthal, Doug Bryan, Brian Renaud, Marie Eguchi, and finally, my family.

Contents

Chapter 1

Introduction: Games, Strategies, and Nash Equilibria

IN WHICH *we define terms like "game" and "strategy" and introduce the problem of generating strategies for games.*

When a human is faced with a new game they digest the rules, consider the motives and capabilities of the other players, and formulate a strategy for playing. This thesis takes steps toward automating that process. A *Strategy Generation Engine* is a system that advises on how to play a game given a formal description of the possible interactions of the players. In the absence of other players, a strategy generation engine is just an optimization routine. That is, it would answer, decision theoretically, the question: What actions maximize my (expected) utility? In the next section, we see how, by way of a cursory introduction to some foundational concepts of game theory, the strategy generation concept can be generalized to the case of multiple agents: a *game*.

1.1 Games of Incomplete Information and Normal-Form Representation

Game theory was founded by von Neumann and Morgenstern [1947] to study situations in which multiple agents (players) interact in order to each maximize an objective (payoff) function determined not only by their own actions but also the actions of other players. Agents always know their own payoff function—i.e., their utility function—but may not know those of the other agents. To capture this, we redefine an agent's payoff to be a function not only of its own and others' actions, but also of *private information*. Thus it is without loss of generality to consider all the payoff functions (or, equivalently, the single multidimensional payoff function) as known by all the agents.

An agent's private information is known more succinctly as its *type*, or sometimes as its *preferences*. The term "type" is used because it is exactly this private information that formally distinguishes the agents. The term "preferences" reflects the fact that the private information determines the agent's utility, given all the agents' actions. For example, in poker an agent's hand constitutes its type. In an auction, the analogous information is how much the agent values the goods being sold.

The agents' types are drawn from a probability distribution, called the *type distribution*. To capture randomness in the game, we introduce a special player, *Nature*, with type dictated, as for any other player, by the type distribution. But Nature has no payoff function and its action is simply its type. In this way Nature could, for example, represent the roll of dice, affecting any or all other players' payoffs. We assume that the possible actions, payoff function, and type distribution are common knowledge.[1] The special case that all agents have exactly one possible type is called complete information. In this thesis, I consider the general case: *games of incomplete information*.

Nash Equilibrium

In the case of a single agent, the optimal policy is straightforward: choose the action that maximizes payoff (utility). If payoff depends on Nature's action, the agent simply maximizes its *expected payoff* given the type distribution. In the case of multiple agents, Nash [1951] proposed a solution concept now known as the *Nash equilibrium* and proved that for finite games (as long as agents can play *mixed strategies*, i.e., randomize among actions) such an equilibrium always exists. A Nash equilibrium that does not employ mixed strategies—called a pure-strategy Nash equilibrium—is a profile of actions such that each action is a *best response* to the rest of the profile. That is, each agent maximizes its own utility given the actions played by the others. The generalization to mixed-strategy equilibria entails maximizing expected utility given the distribution of agent actions.

Although the limitations of the Nash equilibrium as a solution concept have been well studied, particularly the problem of what agents will do in the face of multiple equilibria [van Damme, 1983], finding Nash equilibria remains fundamental to the analysis of games. In fact, for many practical applications, it suffices to find one of many possible Nash equilibria. That equilibrium, by virtue of being the one found by a given solver, becomes *focal* and the equilibrium selection problem is moot. For example, if agent software is distributed that implements an equilibrium policy for a game, no recipient has an incentive to modify the software unless they imagine that others are also modifying it in a certain way, which they would have no incentive to do. Practically, this is reasonable assurance the found equilibrium will actually be played.

Bayes-Nash Equilibrium

The notion of Nash equilibrium needs to be generalized slightly for the case of incomplete information games. We first define an agent's *strategy* as a mapping from its information—private information (type) as well as information revealed during the game—to actions. Another seminal game theorist, Harsanyi[2] [1967], first introduced the concept of agent types and used it to define a *Bayesian game*. A Bayesian game is specified by a set of types T, a set of actions A, a probability distribution F over types, and a payoff function P. Harsanyi defines a *Bayes-Nash equilibrium* (sometimes known as a Bayesian equilibrium) as the simple Nash equilibrium of the non-Bayesian game with set of actions being the set of strategies (functions from T to A), and the payoff function being the expectation of P with respect to F. In this thesis I use "Nash equilibrium" or "equilibrium" interchangeably with "Bayes-Nash equilibrium" when appropriate.

The idea of a normal (or strategic) form representation is to ignore the actions and define the game in terms of the strategies. In other words, per Harsanyi's insight, an agent's set of strategies is equivalent to its set of actions.

Definition 1.1 (Normal-form Game) $\Gamma = \langle n, \{S_i\}, \{u_i()\}\rangle$ *is an* n-player normal-form game, *with* strategy set S_i *the available strategies for player* i *(with typical element* s_i*), and the* payoff

[1] A fact is common knowledge [Fagin *et al.*, 1995] if everyone knows it, everyone knows that everyone knows it, ad infinitum.

[2] Harsanyi, along with Selten, shared the 1994 Nobel Prize in economics with Nash "for their pioneering analysis of equilibria in the theory of non-cooperative games."

function $u_i(s_1, \ldots, s_n)$ *giving the utility accruing to player* i *when players choose the* strategy profile $(s_1, \ldots, s_n)$.

In contrast, *extensive form* is a more compact representation in which payoffs are given for sequences of actions but only implicitly for combinations of agent strategies (mappings from information to actions).

1.2 Symmetric Games

With the exception of the bargaining game in Section 2.4, every game I examine in this thesis is *symmetric*. A game in normal form is symmetric if all agents have the same strategy set, and the payoff to playing a given strategy depends only on the strategies being played, not on who plays them. In other words, a game is symmetric if there are no distinct player roles or identities (aside from Nature). I define this concept formally as follows:

Definition 1.2 (Symmetric Game) *A normal-form game is* symmetric *if the players have identical strategy spaces (*$S_1 = \ldots = S_n = S$*) and* $u_i(s_i, s_{-i}) = u_j(s_j, s_{-j})$*, for* $s_i = s_j$ *and* $s_{-i} = s_{-j}$ *for all* $i, j \in \{1, \ldots, n\}$*.*[3] *Thus we can write* $u(t, s)$ *for the payoff to any player playing strategy* t *when the remaining players play profile* s*. We denote a symmetric game by the tuple* $\langle n, S, u()\rangle$*. (I overload* S *to refer to the number of strategies as well as the set of strategies.)*

A strategy profile with all players playing the same strategy is a *symmetric profile*, or, if such a profile is a Nash equilibrium, a *symmetric equilibrium*.

Many well-known games are symmetric, for example the Prisoners' Dilemma, Chicken, and Rock-Paper-Scissors, as well as standard game-theoretic auction models. Symmetric games may naturally arise from models of automated-agent interactions, since in these environments the agents may possess, by design, identical circumstances, capabilities, and perspectives. But these are encompassed in the agent types so even if they differ it is often accurate to model them as drawn from identical distributions, restoring symmetry. Designers often impose symmetry in artificial environments constructed to test research ideas—for example, the Trading Agent Competition (TAC) [Wellman *et al.*, 2001b] market games—since an objective of the model structure is to facilitate interagent comparisons.

The relevance of symmetry in games stems in large part from the opportunity to exploit this property for computational advantage, as I do throughout this thesis. Symmetry immediately supports more compact representation. The number of distinct profiles in an n-player, S-strategy symmetric game is $\binom{n+S-1}{n}$, as opposed to S^n for the nonsymmetric counterpart. Furthermore, symmetry enables solution methods specific to this structure.

Given a symmetric environment, we typically prefer to identify symmetric equilibria, as asymmetric behavior seems relatively unintuitive [Kreps, 1990], and difficult to explain in a one-shot interaction. Rosenschein and Zlotkin [1994] argue that symmetric equilibria may be especially desirable for automated agents, since programmers can then publish and disseminate strategies for copying, without need for secrecy.

1.3 Summary and Motivation

This thesis concerns the generation and selection of strategies, in particular for trading agents participating in various market mechanisms. Games that trading agents face typically involve private

[3] *s_{-i} denotes the length $n-1$ profile s with the ith element removed.*

information, such as an agent's valuation for a good being bought or sold, and fine-grained action spaces, such as bids of arbitrary amounts of money. Such games are well modeled as infinite games of incomplete information. Generating trading agent strategies means creating a system that can read the description of such a mechanism and output strategies for participating agents.

To make this more concrete, consider an extremely simple auction mechanism, which I use as a running example throughout this thesis: a First-Price Sealed-Bid Auction (FPSB). This is a game in which each agent has one piece of private information: its valuation for an indivisible good being auctioned (in the simplest case we take the common-knowledge type distribution to be i.i.d. $U[0,1]$). Each agent also has a continuum of possible actions: its bid amount. The payoff to an agent is its valuation minus its bid if its bid is highest, and zero otherwise (with ties broken by fair coin toss). We have developed an algorithm that can solve this game. That is, it takes the game description—for the two-player case—and outputs the unique symmetric Bayes-Nash equilibrium. (In this case, for an agent with valuation v, the equilibrium strategy is to bid $v/2$.) Of course, the Nash equilibrium strategy for the particular case of FPSB was identified by auction theorists before computational game solvers existed [Vickrey, 1961; McAfee and McMillan, 1987]. Our algorithm applies to a class of games that includes the above example.

The above method is tractable only for quite simple games. For example, mechanisms that involve iterated bidding and multiple auctions are not likely to be amenable to analytic approaches. For such games, I present an *empirical game methodology* comprising the following broad phases:

1. Generate a small set of candidate strategies. For many domains this must be done semi-manually.

2. Construct via simulation a (partial, approximate) empirical payoff matrix for the simplification of the game restricted to the candidate strategies.

3. Analyze (ideally, solve) the empirical game.

4. Assess the quality of the solutions with respect to the underlying full game.

I discuss approaches for step 1, present techniques for speeding up step 2, compare existing techniques for step 3 in the context of symmetry, and give procedures for achieving step 4. We use small games with known equilibria such as FPSB to test these methods.

We then apply the methodology to two much larger and more realistic market games: Simultaneous Ascending Auctions (SAA), and the Trading Agent Competition (TAC) travel-shopping game [Wellman *et al.*, 2001b]. TAC was created by our research group and first held in 2000. It was held for the sixth time in August 2005 in Edinburgh. The domain involves travel agents shopping for travel packages for a group of hypothetical clients with varying preferences over length of trip, hotel quality, and entertainment options. The shopping involves participating in dozens of simultaneous auctions of various kinds. For example, hotels are sold in multi-unit English ascending auctions while entertainment tickets are bought and sold in continuous double auctions (like the stock market). An agent's payoff is the total utility it achieves for its clients, minus its net expenditure.

SAA is a far simpler model that still captures a core strategic issue in TAC: bidding for complementary goods in concurrent auctions. To apply step 1 to SAA and TAC, I present classes of strategies based on market price prediction. In particular we consider self-confirming price predictions and Walrasian equilibrium prices. Given a set of candidate strategies in SAA or TAC, we apply the subsequent steps of our empirical game methodology to recommend effective strategies in these domains.

1.4 Overview of Thesis

Much of this thesis consists of derivatives of and extensions to published papers with many coauthors. Chapter 2 is based largely on a paper published in 2004 [Reeves and Wellman], with extensions to analyze and discuss additional games amenable to our approach. Chapter 2 is distinct from the rest of the thesis in that it takes an analytic approach to computing strategies in a class of simple, one-shot, infinite games.

Chapter 3 presents our empirical game methodology, the germinal ideas of which we published in the context of market-based scheduling [Wellman *et al.*, 2003c]. We extended the approach later [Reeves *et al.*, 2005], also adding the core ideas for sensitivity analysis, i.e., evaluating the quality of solutions in approximate games. Our approach evolved further in subsequent papers on market-based scheduling, and simultaneous ascending auctions (SAA) in general [MacKie-Mason *et al.*, 2004; Osepayshvili *et al.*, 2005]. We also collected and revised our descriptions of the game solution techniques we employed [Cheng *et al.*, 2004]. That paper presents a collection of theorems about symmetric games which we include here in Appendix C. The player reduction approach to game approximation in Chapter 3 was introduced in the context of TAC and local-effect games [Wellman *et al.*, 2005b]. Other results in Chapter 3, including the proofs of key theorems and many of the experiments with first-price sealed-bid auctions (FPSB) are new to this thesis.

Chapter 4 draws from our work in SAA [MacKie-Mason *et al.*, 2004; Osepayshvili *et al.*, 2005] and TAC [Wellman *et al.*, 2004; Cheng *et al.*, 2005] to present price prediction as a general approach for strategies in games involving simultaneous interdependent auctions. The strategic approaches discussed in Chapter 4 are used to generate the candidate strategies to which we apply our empirical game methodology (Chapter 3) in Chapters 5 and 6.

Chapter 5 coalesces our results from a series of papers on SAA starting with a simple variation on the well-studied straightforward bidding (SB) strategy, called sunk-awareness [Reeves *et al.*, 2005]. We next considered price prediction, showing that even a naive implementation of this approach results in a far superior strategy [MacKie-Mason *et al.*, 2004]. We later generalized the approach to predict price distributions and concluded that a special case of this—self-confirming price distribution prediction—constitutes a robust strategy in a range of SAA environments [Osepayshvili *et al.*, 2005].

In Chapter 6 we use the approach of Chapter 3 in TAC to choose the strategy for our entry in the 2005 competition. We also report on preliminary strategic analysis of TAC in which we derive an equilibrium mixed strategy of shading hotel bids with small probability [Wellman *et al.*, 2003a]. Following the preliminary shading analysis, much of Chapter 6 is based on a preliminary report in which we introduce our approach to choosing our agent strategy for 2005 [Wellman *et al.*, 2005c]. New to this thesis is an analysis of the results of the 2005 competition, vindicating our strategy choice.

Chapter 2

Generating Best-Response Strategies in Infinite Games of Incomplete Information

IN WHICH *we generate strategies for sealed-bid auctions and other one-shot, two-player, infinite games of incomplete information.*

In a *one-shot game*[1] an agent immediately receives a payoff after choosing a single action based only on knowledge of its own type, the payoff function, and the distribution from which types are drawn. By *infinite game*, we mean that the action spaces are continuous. In this chapter we consider one-shot, two-player, infinite games of incomplete information. Furthermore, we restrict agent types to a subset of the reals. A strategy in this context is a one-dimensional function ($\mathbb{R} \to \mathbb{R}$) from the set of types to the set of actions. The case where the set of types and actions are finite (and especially when there are no types—complete information) has been well studied in the computational game theory literature. In the next section we discuss the state-of-the-art finite game solver, GAMBIT. But for incomplete information games with a continuum of actions available to the agents we know of no available algorithms, though many particular infinite games of incomplete information have been solved in the literature.

In this chapter we define a broad class of games and present an algorithm for computing exact best-response strategies.[2] The definition of Nash equilibrium (each agent playing a best response to the other strategies) invites an obvious algorithm for solving a game: start with a profile of seed strategies and iteratively compute best-response profiles until a fixed-point is reached. This process (when it converges) yields a profile that is a best response to itself and thus a Nash equilibrium. After presenting our class of games (Section 2.2) and our algorithm (Section 2.3) we show examples of iterating the best-response computation to automatically solve various new and existing games (Section 2.4).

2.1 Finite Game Approximations

GAMBIT [McKelvey *et al.*, 1992] is a software package incorporating several algorithms for solving finite games [McKelvey and McLennan, 1996]. The original 1992 implementation of GAMBIT

[1] Fudenberg and Tirole [1991] call them *one-stage games*.

[2] The solutions are exact as long as the inputs are rational since all calculations are done in the rational field.

was limited to normal-form games (Definition 1.1). GALA [Koller and Pfeffer, 1997] introduced constructs for schematic specification of extensive-form games, with a solver exploiting recent algorithmic advances [Koller *et al.*, 1996]. GAMBIT subsequently incorporated these GALA features, and currently stands as the standard in general finite game solvers.

We employ a standard first-price sealed-bid auction (FPSB) to compare our approach for the full infinite game to a discretized version amenable to finite game solvers. Consider a discretization of (FPSB) with two players, nine types, and nine actions. Both players have a randomly (uniform) determined valuation (t) from the set $\{0,\ldots,8\}$ and the available actions (a) are to bid an integer in the set $\{0,\ldots,8\}$. Let $[p,a;a']$ denote the mixed strategy of performing action a with probability p, and a' with probability $1-p$. (There happened never to be mixed strategies with more than two actions.) GAMBIT solves this game exactly, finding the following Bayes-Nash equilibrium:

$t:$	0	1	2	3	4	5	6	7	8
$a(t):$	0	0	1	1	2	$[.455,2;3]$	3	3	$[.727,3;4]$

This result is indeed close to the unique symmetric Bayes-Nash equilibrium, $a(t)=t/2$, of the corresponding infinite game (see Section 2.4) despite varying asymmetrically—an artifact of the discretization. However, the calculation took 90 minutes of cpu time and 17MB of memory.[3] When 2 additional types and actions are added to the discretization, a similar equilibrium results, requiring 23 hours of cpu time and 34MB of memory.

GAMBIT's algorithm [Koller *et al.*, 1996] is worst-case exponential in the size of the game tree which is itself size $O(n^4)$ in the size of the type/action spaces. Based on this complexity and our timing results, we conclude that (regardless of hardware advances) we are near the limit of what GAMBIT's algorithm can compute. In Section 2.4 we show how our method immediately finds the solution to the full infinite FPSB game. And in Chapter 3, we present better approaches to discrete approximations, for when it is necessary to employ them.

2.2 Infinite Games and Bayes-Nash Equilibria

We consider a class of two-player games, defined by a payoff structure that is analytically restrictive, yet captures many well-known games of interest. Let t denote the subject agent's type and a its action, and t' and a' the type and action of the other agent.[4] We assume that types are scalars drawn from piecewise-uniform probability distributions, and payoff functions take the following form:

$$u(t,a,t',a')=\begin{cases}\theta_1 t+\rho_1 a+\theta_1' t'+\rho_1' a'+\phi_1 & \text{if } -\infty < a+\alpha a' < \beta_2\\ \theta_2 t+\rho_2 a+\theta_2' t'+\rho_2' a'+\phi_2 & \text{if } \beta_2 \le a+\alpha a' \le \beta_3\\ \cdots & \\ \theta_{I-1} t+\rho_{I-1} a+\theta_{I-1}' t'+\rho_{I-1}' a'+\phi_{I-1} & \text{if } \beta_{I-1} < a+\alpha a' < \beta_I\\ \theta_I t+\rho_I a+\theta_I' t'+\rho_I' a'+\phi_I & \text{if } \beta_I \le a+\alpha a' \le +\infty.\end{cases} \tag{2.1}$$

Our class comprises games with payoffs that are linear[5] functions from own type and action, conditional on a linear comparison between own and other agent action. The form is parameterized

[3]The machine had four 450MHz Pentium 2 processors and 2.5GB RAM, running Linux kernel 2.4.18.

[4]Following convention, we often call the other agent the "opponent" even though the term is less apt for non-zero-sum games.

[5]Functions with constant terms are technically *affine* rather than linear but we ignore that distinction from here on.

by α, β_i, θ_i, ρ_i, θ'_i, ρ'_i, and ϕ_i, where $i \in \{1, \dots, I\}$ indexes the comparison case. (We define $\beta_1 \equiv -\infty$ and $\beta_{I+1} \equiv +\infty$ for notational convenience in our algorithm description below.) The β_i, β_{i+1} regions alternate between open and closed intervals as this is without loss of generality for arbitrary specification of boundary types ('$<$' vs. '$\leq$') on the regions[6] and in particular allows the implementation of common tie-breaking rules for sealed-bid auctions.

This parameterized payoff function captures many known mechanisms. Table 2.1 shows the parameter settings for many such games, plus new ones we introduce in Section 2.4. Given a game description in this form, we search for Bayes-Nash equilibria through a straightforward iterative process. Starting with a seed strategy profile (typically based on a myopic or naive strategy such as truthful bidding), we repeatedly compute best-response profiles until reaching a fixed-point or cycle. A strategy profile that is a best response to itself is, by definition, a Bayes-Nash equilibrium. We show that this process is effective at finding equilibria for certain games in our class. For all games in our class, the best-response algorithm can be used to verify candidate equilibria or ε-equilibria found by alternate means.

Our method considers only pure strategies. Although mixed strategies are generally required for infinite as well as finite games, there are broad classes of infinite games for which pure-strategy equilibria are known to exist. For example, Debreu [1952] shows that equilibria in pure strategies exist for infinite games of complete information with action spaces that are compact, convex subsets of a Euclidean space $\mathbb{R}^n$, and payoffs that are continuous and quasiconcave in the actions. However, games in our class may have discontinuous payoff functions (most auctions do). Furthermore, our action space—being the set of piecewise linear functions on the unit interval—is not a compact subset of $\mathbb{R}^n$. Athey [2001] proves the existence of pure-strategy Nash equilibria for games of incomplete information satisfying a property called the single-crossing condition (SCC). These results encompass many familiar games of economic relevance, including auction games such as FPSB. Our class includes games violating SCC, for which search in the space of pure strategies may not be sufficient. Nevertheless, an ability to compute best responses for the broadest possible class of games is useful in itself.

The best-response algorithm takes as input a piecewise linear strategy with K pieces ($K - 1$ piece boundaries),

$$s(t) = \begin{cases} m_1 t + b_1 & \text{if } -\infty < t \leq c_2 \\ m_2 t + b_2 & \text{if } c_2 < t \leq c_3 \\ \dots & \\ m_{K-1} t + b_{K-1} & \text{if } c_{K-1} < t \leq c_K \\ m_K t + b_K & \text{if } c_K < t \leq +\infty, \end{cases} \tag{2.2}$$

represented by the vectors $\boldsymbol{c}$, $\boldsymbol{m}$, and $\boldsymbol{b}$. The piecewise-linear strategy class is sufficiently flexible to approximate any strategy, although of course the complexity of the strategy or quality of the approximation suffers as nonlinearity increases.

A two-player game is symmetric (Definition 1.2) if both players face the same payoff function. For symmetric games we start with a single seed strategy, to be repeatedly replaced with the strategy that responds best to it.[7] Most of the examples presented in this chapter are symmetric and have symmetric pure equilibria. For asymmetric games, we start with a pair of seed strategies, on every

[6] For example, to specify u_1 in $(-\infty, a]$, u_2 in $(a, b]$, u_3 in (b, c), u_4 at c, and u_5 in (c, ∞), translate to the alternating open/closed specification: u_1 in $(-\infty, a - \varepsilon)$, u_1 in $[a - \varepsilon, a]$, u_2 in (a, b), u_2 in $[b, b]$, u_3 in (b, c), u_4 in $[c, c]$, u_5 in (c, ∞).

[7] It can be shown that symmetric games must have symmetric equilibria, although there are some symmetric games with only asymmetric *pure* equilibria (see Appendix C).

	Game	$\boldsymbol{\theta}$	$\boldsymbol{\rho}$	$\boldsymbol{\theta'}$	$\boldsymbol{\rho'}$	$\boldsymbol{\phi}$	$\boldsymbol{\beta}$	α
	FPSB	$0, 1/2, 1$	$0, -1/2, -1$	$0, 0, 0$	$0, 0, 0$	$0, 0, 0$	$0, 0$	-1
	Vickrey Auction	$0, 1/2, 1$	$0, 0, 0$	$0, 0, 0$	$0, -1/2, -1$	$0, 0, 0$	$0, 0$	-1
	Vicious Vickrey Auction	$0, \frac{1-k}{2}, 1-k$	$k, k/2, 0$	$-k, -k/2, 0$	$0, \frac{k-1}{2}, k-1$	$0, 0, 0$	$0, 0$	-1
	Supply Chain Game	$-1, -1, 0$	$1, 1, 0$	$0, 0, 0$	$0, 0, 0$	$0, 0, 0$	v, v	1
	Bargaining Game (seller)	$-1, -1, 0$	$1-k, 1-k, 0$	$0, 0, 0$	$k, k, 0$	$0, 0, 0$	$0, 0$	-1
	(buyer)	$0, 1, 1$	$0, -k, -k$	$0, 0, 0$	$0, 1-k, 1-k$	$0, 0, 0$	$0, 0$	-1
	All-Pay Auction	$0, 1/2, 1$	$-1, -1, -1$	$0, 0, 0$	$0, 0, 0$	$0, 0, 0$	$0, 0$	-1
	War of Attrition	$0, 1/2, 1$	$-1, -1/2, 0$	$0, 0, 0$	$0, -1/2, -1$	$0, 0, 0$	$0, 0$	-1
*	Shared-Good Auction	$0, 1/2, 1$	$0, -1/4, -1/2$	$0, 0, 0$	$1/2, 1/4, 0$	$0, 0, 0$	$0, 0$	-1
*	Joint Purchase Auction	$0, 1$	$0, -1/2$	$0, 0$	$0, 1/2$	$0, -C/2$	C	1
	Subscription Game	$0, 1$	$0, -1$	$0, 0$	$0, 0$	$0, 0$	C	1
	Contribution Game	$0, 1$	$-1, -1$	$0, 0$	$0, 0$	$0, 0$	C	1

Table 2.1: Various mechanisms as special cases of the parameterized payoff function in Equation 2.1. Starred games are newly defined in this thesis. Note that the bargaining game, being asymmetric, is described by two payoff functions. These games are discussed in Section 2.4.

iteration computing a best response to each to get the new pair. We may also find asymmetric equilibria when iterating from a single strategy (equivalently, a symmetric profile). In this case, a cycle of length two (given our restriction to two-player games, but regardless of whether the game is symmetric) constitutes an asymmetric equilibrium.

2.3 Existence and Computation of Piecewise Linear Best-Response Strategies

Here we present our algorithm to compute the best response to a given strategy by way of a constructive proof that in our class of games, best responses to piecewise linear strategies are themselves piecewise linear. Intuitively, the proof proceeds by first deriving an algebraic expression for expected utility against the given strategy in terms of the payoff parameters, the distribution parameters, the opponent strategy parameters, own type, and own action. By appropriate partitioning of the action space, the expected utility is expressed as a piecewise polynomial in the agent's action. We then show that the action maximizing that expression (the best response) is a piecewise linear expression of the agent's type. Finally, we establish a bound for the number of pieces in the best-response strategy.

Theorem 2.1 *Given a payoff function with I regions as in Equation 2.1, an opponent type distribution with cdf F that is piecewise uniform with J pieces and $J-1$ piece boundaries $\{d_2, \ldots, d_J\}$, and a piecewise linear strategy function with K pieces as in Equation 2.2, the best-response strategy is itself a piecewise linear function with no more than $2(I-1)(J+K-2)$ piece boundaries.*

Proof. Finding the best response strategy means maximizing expected utility over the other agent's type distribution. Let T be the random variable denoting the other agent's type.

First, redefine $s(t)$ to include additional redundant boundary points $\{d_2, \ldots, d_J\}$ so there are now $J+K-2$ boundary points of $s(t)$, $\{c_2, \ldots, c_{J+K-1}\}$, and

$$s(t) = \begin{cases} m_1 t + b_1 & \text{if } -\infty < t \leq c_2 \\ \ldots & \\ m_{J+K-1} t + b_{J+K-1} & \text{if } c_{J+K-1} < t \leq +\infty. \end{cases}$$

We now express the expected utility, factored over the pieces of $s()$ and $u()$, as

$$\begin{aligned} EU(t,a) = E_T[u(t,a,T,s(T))] = \\ \sum_{i=1}^{I} \sum_{j=1}^{J+K-1} E\Big[(\theta_i t + \rho_i a + \theta_i' T + \rho_i'(m_j T + b_j) + \phi_i \ \Big| \\ c_j < T \leq c_{j+1},\ \beta_i \lessdot a + \alpha(m_j T + b_j) \lessdot \beta_{i+1}\Big] \\ \cdot \Pr(c_j < T \leq c_{j+1},\ \beta_i \lessdot a + \alpha(m_j T + b_j) \lessdot \beta_{i+1}). \end{aligned}$$

(We use the notation "$x_i \lessdot y$" to denote $x_i < y$ if i is odd and $x_i \leq y$ if i is even.)

If $\alpha m_j = 0$ then the summand reduces to

$$\begin{cases} (\theta_i t + \rho_i a + (\theta_i' + \rho_i' m_j)\dfrac{c_j + c_{j+1}}{2} + \rho_i' b_j + \phi_i) & \\ \quad \cdot (F(c_{j+1}) - F(c_j)) & \text{if } \beta_i - \alpha b_j \lessdot a \lessdot \beta_{i+1} - \alpha b_j \\ 0 & \text{otherwise.} \end{cases}$$

(The derivation of the cdf $F()$ of a piecewise uniform distribution is in Appendix A.1.)

For the case of $\alpha m_j \neq 0$, first define

$$x_{ij}(a) \equiv \frac{\beta_i - \alpha b_j - a}{\alpha m_j} \quad \text{and}$$
$$y_{ij}(a) \equiv \frac{\beta_{i+1} - \alpha b_j - a}{\alpha m_j}$$

with x and y swapped if $\alpha m_j < 0$. We also introduce $mm(a,b,x) \equiv min(b, max(a,x))$.

We consider first the probability term in the summand, rewriting it as

$$\begin{aligned} p_{ij}(a) \equiv& \Pr\left(\beta_i \underset{\cdot}{<} a + \alpha \cdot (m_j T + b_j) \underset{\cdot}{<} \beta_{i+1} \ \&\ c_j < T \leq c_{j+1}\right) \\ =& \Pr\left(x_{ij}(a) \underset{\cdot}{<} T \underset{\cdot}{<} y_{ij}(a) \ \&\ c_j < T \leq c_{j+1}\right) \\ =& F(mm(c_j, c_{j+1}, y_{ij}(a))) - F(mm(c_j, c_{j+1}, x_{ij}(a))). \end{aligned}$$

For the expectation term in the summand, we first define

$$\begin{aligned} \overline{xy}_{ij}(a) \equiv& E[T \mid mm(c_j, c_{j+1}, x_{ij}(a)) \underset{\cdot}{<} T \underset{\cdot}{<} mm(c_j, c_{j+1}, y_{ij}(a))] \\ =& \frac{mm(c_j, c_{j+1}, x_{ij}(a)) + mm(c_j, c_{j+1}, y_{ij}(a))}{2}. \end{aligned}$$

We can now express the expected utility, $EU(t,a)$, as

$$\sum_{i=1}^{I} \sum_{j=1}^{J+K-1} (\theta_i t + \rho_i a + (\theta_i' + \rho_i' m_j)\overline{xy}_{ij}(a) + \rho_i' b_j + \phi_i) \cdot p_{ij}(a). \tag{2.3}$$

This expression is a piecewise second degree polynomial in a and simply linear in t. Treating it as a function of a, parameterized by t, we can find the boundaries for the polynomial pieces (which will be expressions of t). This is done by setting the arguments of the maxes and mins equal and solving for a, yielding the following four action boundaries for each region $\{\beta_i, \beta_{i+1}\}$ in $u()$ and each region $\{c_j, c_{j+1}\}$ in $s()$:

$$\begin{aligned} c_{j+1} = y_{ij}(a) \quad &\Longrightarrow \quad a = \beta_{i+1} - \alpha \cdot (m_j c_{j+1} + b_j) \\ c_j = y_{ij}(a) \quad &\Longrightarrow \quad a = \beta_{i+1} - \alpha \cdot (m_j c_j + b_j) \\ c_{j+1} = x_{ij}(a) \quad &\Longrightarrow \quad a = \beta_i - \alpha \cdot (m_j c_{j+1} + b_j) \\ c_j = x_{ij}(a) \quad &\Longrightarrow \quad a = \beta_i - \alpha \cdot (m_j c_j + b_j). \end{aligned}$$

This yields a total of at most $2(I-1)(J+K-2)$ unique action boundaries. So expected utility is now expressible as a piecewise polynomial in a (parameterized by t) with at most $2(I-1)(J+K-2)+1$ pieces.

For arbitrary t, we can find the action a that maximizes $EU(t,a)$ by evaluating at each of the boundaries above and wherever the derivative (of each piece) with respect to a is zero. This yields up to $2(I-1)(J+K-2)+1$ critical points, all simple linear functions of t. Call this set of candidate actions C and the corresponding set of expected utilities $EU(t,C)$. The best-response function can then be expressed, for given t, as $\arg\max_C(EU(t,C))$. This is a *piecewise max* of the linear functions in C, and so it is piecewise linear.

It remains to establish an upper bound on the resulting number of distinct ranges for t. We claim the size of C, $2(I-1)(J+K-2)+1$, is such an upper bound. To see this, first note that the piecewise max of a set of linear functions must be convex (since, inductively, the max of a line and convex function is convex). It is now sufficient to show that at most one new t range can be added by taking the max of a linear function of t and a piecewise linear convex function of t. Suppose the opposite, that the addition of one line adds two pieces. They cannot be contiguous else they would be one piece. So there must be a piece of the convex function between the two pieces of the line. This means the convex function goes below, then above, then below the line and this violates convexity. Therefore, each line in C adds at most one piece to the piecewise max of C and therefore the piecewise linear best response to $s()$ has at most $2(I-1)(J+K-2)+1$ pieces and thus $2(I-1)(J+K-2)$ type boundaries. □

Our algorithm for finding a best response follows this constructive proof. Finding C takes time $O(IJK)$. To actually find the piecewise linear function, $\arg\max_C(EU(t,C))$, we employ a brute force approach that requires $O((IJK)^2)$ time. First, we find all possible piece boundaries by taking all pairs in C, setting them equal, and solving for t. For each t range we then compute $\arg\max_C(EU(t,C))$ and merge whenever contiguous ranges have the same argmax. As the proof shows, this will yield at most $2(I-1)(J+K-2)$ type boundaries. Thus, we have shown how to find the piecewise linear best response to a piecewise linear strategy in polynomial time. The resulting function is converted to the same strategy representation ($\boldsymbol{c}$, $\boldsymbol{m}$, $\boldsymbol{b}$) that the algorithm takes as a seed for the opponent strategy.

2.4 Examples

Here we consider existing and new games and show that our method for finding best responses can confirm or rediscover known results as well as find equilibria in new games. Refer to Table 2.1 on page 10 for how these and other games are encoded per Equation 2.1.

There are many games not analyzed here to which our approach is amenable, such as the All-Pay auction (both winner and loser pay their bids; can be used to model activities like lobbying), the War of Attrition (both winner and loser pay the second-highest price) [Krishna and Morgan, 1997], incomplete information versions of Cournot or Bertrand games [Fudenberg and Tirole, 1991, p. 215], and various voluntary participation games in which agents choose an amount to contribute for a joint good and receive utility based on the sum of the contributions. There is a large body of literature on such games, also referred to as private provision of public goods. In one variant, *the subscription game*, agents specify their contributions but their money is refunded if the sum is not sufficient to acquire the good. The *contribution game* is the same but with no refund. Until Menezes *et al.* [2001], only complete information versions of these games were considered in the literature. We additionally define in Table 2.1 a new variant called the *joint purchase auction* which is like the subscription game except that any surplus money collected is split evenly among the participants.[8] We do not analyze this game here but have found that it has a Nash equilibrium similar to that of the bargaining game analyzed in Section 2.4.

Our approach is not needed for incentive compatible mechanisms such as the Vickrey auction, but, reassuringly, our algorithm returns the dominant strategy of truthful bidding [Vickrey, 1961] as a best response to any other strategy in that domain.

[8] An arguably fairer version of the game in which the surplus is divided proportionally to the contributions is unfortunately not in the class of games defined by Equation 2.1.

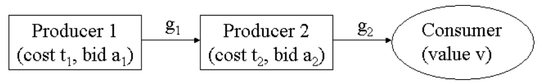

Figure 2.1: Supply Chain game with two producers in series.

First-Price Sealed-Bid Auction

We consider the first-price sealed-bid auction (FPSB) with types that are drawn from $U[0, 1]$ and the following payoff function:

$$u(t, a, a') = \begin{cases} t - a & \text{if } a > a' \\ \dfrac{t-a}{2} & \text{if } a = a' \\ 0 & \text{otherwise.} \end{cases} \tag{2.4}$$

In words, two agents have private valuations for a good and they submit sealed bids expressing their willingness to pay. The agent with the higher bid wins the good and pays its bid, thus receiving a surplus of its valuation minus its bid. The losing agent gets zero payoff. In the case of a tie, a winner is chosen randomly, so the expected utility is the average of the winning and losing utility.

This game can be given to our solver by setting the payoff parameters as in Table 2.1. The algorithm also needs a seed strategy, for which we can use the default strategy of truthful bidding (always bidding one's true valuation: $a(t) = t$ for $t \in [0, 1]$). This strategy is encoded as $\boldsymbol{c} = \langle 0, 1 \rangle$, $\boldsymbol{m} = \langle 0, 1, 0 \rangle$, and $\boldsymbol{b} = \langle 0, 0, 0 \rangle$ per Equation 2.2. Note that the first and last elements of $\boldsymbol{m}$ and $\boldsymbol{b}$ are irrelevant as they correspond to type ranges that occur with zero probability. After a single iteration (a fraction of a second of cpu time), our solver returns the strategy $a(t) = t/2$ for $t \in [0, 1]$ which is the known Bayes-Nash equilibrium for this game (see Theorem 3.1). We find that in fact we reach this fixed point in one or two iterations from a variety of seed strategies—specifically, strategies $a(t) = mt$ for $m > 0$. We approach the fixed point asymptotically (within 0.001 in ten iterations) for seed strategies $a(t) = mt + b$ with $b > 0$.

Supply-Chain Game

This example derives from our work in mechanisms for supply chain formation [Walsh *et al.*, 2000; Walsh, 2001, Chapter 6]. Consider a supply chain with two producers in series, and one consumer (see Figure 2.1). Producer 1 has output g_1 and no input. Producer 2 has input g_1 and output g_2. The consumer—which is not an agent in this model—wants good g_2. The producer costs, t_1 and t_2, are chosen randomly from $U[0, 1]$. A producer knows its own cost with certainty, but not the other producer's cost—only the distribution (which is common knowledge). The consumer's value, $v \geq 1$, for good g_2 is also common knowledge.

The producers place bids a_1 and a_2. If $a_1 + a_2 \leq v$, then all agents win their bids in the auction and the surplus of producer i is $a_i - t_i$. Otherwise, all agents receive zero surplus. In other words, the two producers each ask for a portion of the available surplus, v, and get what they ask minus their costs if the sum of their bids is less than v.[9]

[9]Thus the expected efficiency obtained with any set of bidding policies is equal to the probability that the solution is computed by the auction.

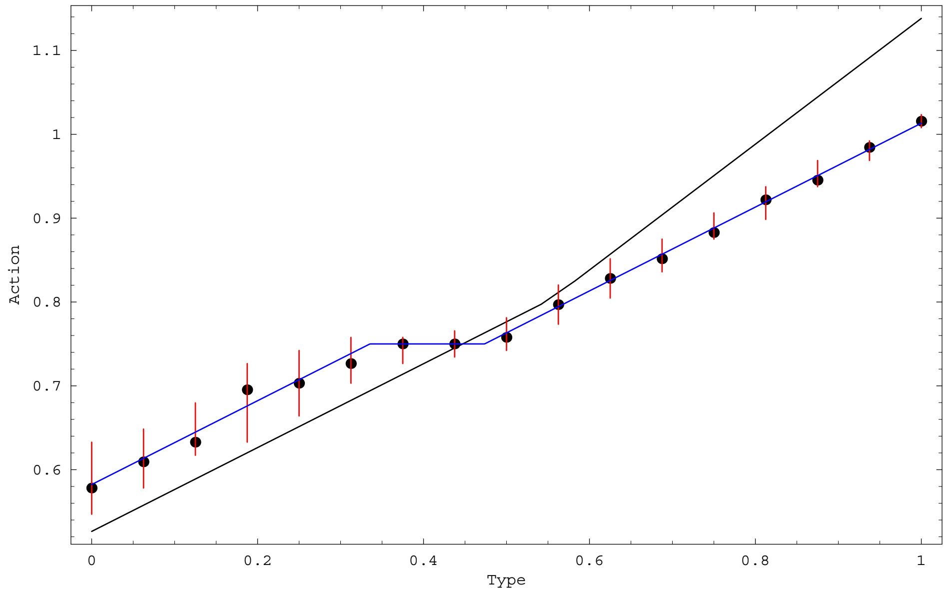

Figure 2.2: Hand-coded strategy for the Supply Chain game of Section 2.4, along with best response and empirical verification of best response (the error bars for the empirically estimated strategy are explained in Appendix B).

Walsh *et al.* [2000] propose a strategy for supply-chain games defined on general graphs. In the more general setting, it is the best known strategy (for lack of any other proposed strategies in the literature). For the particular instance of Figure 2.1, the strategy works out to:

$$a(t) = \begin{cases} t/2 + (v/2 - 1/4) & \text{if } 0 \le t < v - 1 \\ 3t/4 + v/4 & \text{otherwise.} \end{cases}$$

Our best-response finder proves that this strategy is not a Nash equilibrium and shows how to optimally exploit agents who are playing it.

Figure 2.2 shows this strategy for the game with $v = (10 - \sqrt{5})/5 \approx 1.55$ (chosen so that there is a 0.9 probability of positive available surplus) along with the best response, as determined by our algorithm and confirmed by Monte Carlo simulation.

When we perform further best-response iterations it eventually falls into a cycle of period two consisting of the following strategies (where $x = 3/4$):

$$a_1(t_1) = \begin{cases} x & \text{if } t_1 \le x \\ v & \text{otherwise} \end{cases} \tag{2.5}$$

$$a_2(t_2) = \begin{cases} v - x & \text{if } t_2 \le v - x \\ v & \text{otherwise.} \end{cases} \tag{2.6}$$

The following theorem confirms that we have found an equilibrium, and follows an analogous result [Nash, 1953] for the similar (complete information) *Nash demand game*.

Theorem 2.2 *Equations 2.5 and 2.6 constitute an asymmetric Bayes-Nash equilibrium for the supply-chain game, for any* $x \in [0, v]$.

Proof. Assume producer 2 bids according to Equation 2.6. Since producer 1 cannot improve its chance of winning with a bid below x, and can never win with a bid above x, producer 1 effectively has the choice of winning with a bid of x or getting nothing. Producer 1 would choose to win at x precisely when $t_1 \leq x$. Hence, (2.5) is a best response by producer 1. By a similar argument, (2.6) is a best response by producer 2, if producer 1 follows Equation 2.5. □

Following is a more interesting equilibrium, which our solver did *not* find but we were able to derive manually and our best-response finder confirms.

Theorem 2.3 *When $v \in [3/2, 3]$, the following strategy is a symmetric Bayes-Nash equilibrium for the Supply Chain game:*

$$a(t) = \begin{cases} 2v/3 - 1/2 & \textit{if } t \leq 2v/3 - 1 \\ t/2 + v/3 & \textit{otherwise.} \end{cases}$$

Appendix A.2 contains the proof which is essentially an application of our best-response algorithm to the particular game and strategy above. When this strategy is used as the seed strategy for our solver with any particular v, the same strategy is output, thus confirming that it is a Bayes-Nash equilibrium.

Bargaining Game

The supply chain game is similar to a two-player sealed-bid double auction, or bargaining game. In this game there is a buyer with value v' and a seller with cost c', each drawn from distributions that are common knowledge. The buyer and seller place bids and if the buyer's is greater than the seller's, they exchange the good at a price that is some linear combination of the two bids. In the supply-chain example, we can model the seller as producer 1, with $c' = t_1$. Because the consumer reports its true value, which is common knowledge, we can model the buyer as the combination of the consumer and producer 2, with $v' = v - t_2$. However, to make the double auction game isomorphic to our supply-chain example, we need to alter the game so that excess surplus is thrown away instead of shared. The bargaining game as defined above has been well studied in the literature [Chatterjee and Samuelson, 1983; Leininger *et al.*, 1989; Satterthwaite and Williams, 1989]. We consider the special case of the bargaining game where the sale price is halfway between the buy and sell offers, and the valuations are $U[0, 1]$. The payoff function for this game is encoded in Table 2.1.

The following is a known equilibrium [Chatterjee and Samuelson, 1983] for a seller (1) and buyer (2):

$$a_1(t_1) = 2t_1/3 + 1/4$$
$$a_2(t_2) = 2t_2/3 + 1/12.$$

Our solver finds this equilibrium after several iterations (with tolerance 0.001) when seeded with truthful bidding.

Shared-Good Auction

Consider two agents who jointly own an inherently unsharable good and seek a mechanism to decide who should buy the other out and at what price. (For example, two roommates could use this

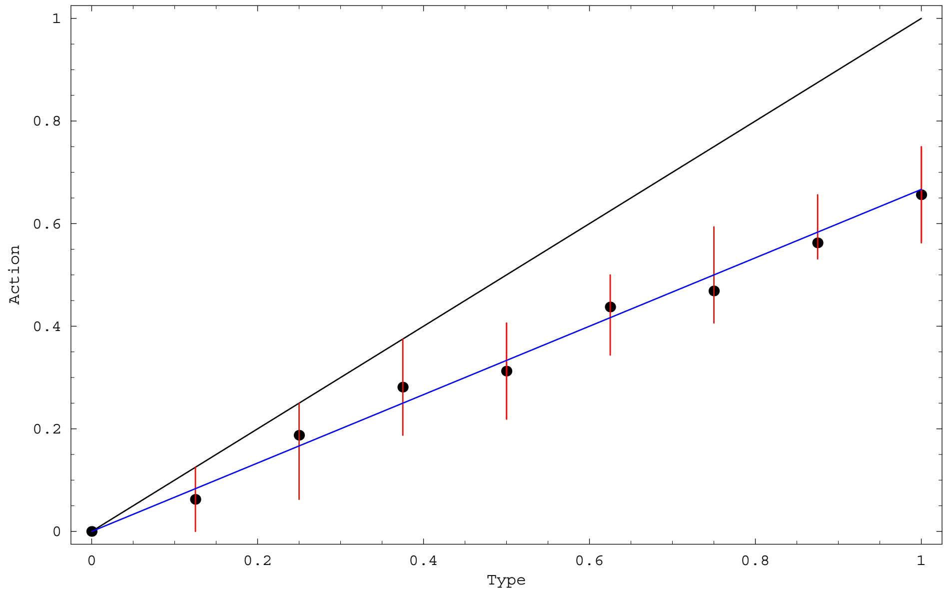

Figure 2.3: $a(t) = 2t/3$ is a best response to truthful bidding in the shared-good auction with $[A, B] = [0, 1]$. This strategy is in turn a best response to itself, thus confirming the equilibrium.

mechanism to decide who gets the better bedroom.[10]) Assume that it is common knowledge that the agents' valuations (types) are drawn from $U[A, B]$. We propose the mechanism

$$u(t, a, a') = \begin{cases} t - a/2 & \text{if } a > a' \\ a'/2 & \text{otherwise} \end{cases}$$

which we chose because it has the property that if players bid their true valuations, the mechanism would allocate the good to the agent who valued it most and split the surplus evenly between the two (each agent would get a payoff of $t/2$ where t is the larger of the two valuations). The following Bayes-Nash equilibrium also allocates the good to the agent who values it most, but that agent gets up to twice as much surplus as the other agent, depending on the minimum valuation A.

Theorem 2.4 *The following is a Bayes-Nash equilibrium for the shared-good auction game when valuations are drawn from $U[A, B]$:*

$$a(t) = \frac{2t + A}{3}.$$

Appendix A.3 contains the proof.

Our solver finds this equilibrium exactly (for any specific $[A, B]$) in one iteration from truthful bidding. We confirm the result via simulation as shown in Figure 2.3.

[10]Thanks to Kevin Lochner who both inspired the need for and helped define this mechanism.

Vicious Vickrey Auction

Brandt and Weiß [2001] introduce the following auction game:

$$u(t,a,t',a') = \begin{cases} (1-k)(t-a') & \text{if } a > a' \\ \dfrac{(1-k)(t-a') - k(t'-a)}{2} & \text{if } a = a' \\ -k(t'-a) & \text{otherwise.} \end{cases} \tag{2.7}$$

It is a Vickrey auction generalized by the parameter k which allows agents to be *spiteful* in the sense of getting disutility from the other agent's utility. For example, this might be the case for businesses that are competitors.

Brandt and Weiß derive an equilibrium for a complete information version of this game. Our game solver can address the incomplete information setting.

Theorem 2.5 *The following is a Bayes-Nash equilibrium for the Vicious Vickrey auction game when valuations are drawn from $U[0,1]$:*

$$a(t) = \frac{k+t}{k+1}.$$

Appendix A.4 contains the proof. Our solver finds this equilibrium (for various specific values of k) within several iterations from a variety of seed strategies.

In recent work, Morgan *et al.* [2003] and Brandt *et al.* [2005] have independently (from each other as well as from this work) investigated incomplete-information versions of the Vicious Vickrey auction. Morgan *et al.* as well as Brandt *et al.* report the equilibrium for Vicious Vickrey given in Theorem 2.5, and in fact do so for the more general case of arbitrary type distributions and number of agents. Morgan *et al.* use a payoff function rescaled from (2.7) by a factor of $\frac{1}{1-k}$ and with a slightly different parameter, α, which is $\frac{k}{1-k}$ in our formulation. Interestingly, the equilibrium does not depend on the number of agents.

2.5 Related Work

The seminal works on game theory are von Neumann and Morgenstern [1947] and Nash [1951]. There are several modern general texts [Aumann and Hart, 1992; Fudenberg and Tirole, 1991; Mas-Colell *et al.*, 1995] that analyze many of the games in Section 2.4. Algorithms for solving finite games include the classic Lemke-Howson algorithm [Lemke and Howson, Jr., 1964] for solving bimatrix games (two-agent finite games of complete information). In addition to the algorithms discussed in connection with GAMBIT (Section 2.1), there has been recent work [La Mura, 2000; Kearns *et al.*, 2001; Bhat and Leyton-Brown, 2004] in algorithms for computing Nash equilibria in finite games by exploiting compact representations. Govindan and Wilson [2003, 2002] have recently found new algorithms for searching for equilibria in normal-form and extensive-form games that are faster than any algorithm implemented in GAMBIT. Blum *et al.* [2003] have extended and implemented these algorithms in a package called GAMETRACER. Singh *et al.* [2004] adapt graphical-game algorithms for the incomplete information case, including a class of games with continuous type ranges and discrete actions.

The approach of finding Nash equilibria by iterated best-response, sometimes termed *best-reply dynamics*, dates back to Cournot [1838]. A similar approach known as *fictitious play* was introduced by Robinson [1951] and Brown [1951] in the early days of modern game theory. Fictitious play employs a best response, not to the single strategy from the last iteration, but a composite strategy formed by mixing the strategies encountered in previous iterations according to their historical frequency. This method generally has better convergence properties than best-response, but Shapley [1964] showed that fictitious play need not converge in general. Milgrom and Roberts [1991] cast both of these iterative methods as special cases of what they term *adaptive learning* and show that in a class of games of complete information, all adaptive learning methods converge to the unique Nash equilibrium. Fudenberg and Levine [1998] provide a good general text on iterative solution methods (i.e., learning) for finite games. Hon-Snir *et al.* [1998] apply this approach to a particular auction game with complete information. The *relaxation algorithm* [Uryasev and Rubinstein, 1994], applicable to infinite games, but only complete information games, is a generalization of best-response dynamics that has been shown to converge for some classes of games. Finally, Turocy [2001] presents a Monte Carlo "stochastic approximation" approach to estimating best responses in a very broad class of auction games and proposes iterating the procedure to find equilibria.

The literature is rife with examples of analytically computed equilibria for particular auction games. For example, Milgrom and Weber [1982] derive equilibria for first- and second-price auctions with affiliated signals, Gordy [1998] finds closed-form equilibria in certain common-value auctions given particular signal distributions, and Menezes *et al.* [2001] find equilibria for the subscription and contribution games. Additionally, as discussed in Section 2.4, Brandt *et al.* [2005] and Morgan *et al.* [2003] have independently derived the same symmetric Bayes-Nash equilibrium for the more general case of the incomplete-information Vicious Vickrey auction with arbitrary type distribution and number of agents.

Chapter 3

Empirical Game Methodology

IN WHICH *we lay out a collection of techniques for generating strategies in games too big to solve exactly, and show experimental evidence that they work.*

The previous chapter describes a class of games for which we can find best-response strategies and in many cases Nash equilibria. The strategy generation in that case is fully automated, yielding a profile of strategies from only the description of the game. But the solution methods described in Chapter 2 will simply not scale to games with several players with multidimensional types where information is revealed dynamically during the game. Consider, for example, bidding in concurrent auctions for related goods (a problem I address in detail in subsequent chapters). Such a game would be defined by (1) a distribution over agents' value functions defined over all subsets of a set of goods, (2) a strategy space mapping valuations crossed with all possible quote histories to a bid vector for the auctions, and (3) a payoff function incorporating the auction rules that assigns payoffs based on valuations and bidding actions. The game size is astronomical with no game-theoretic solution in sight.[1]

There are many realistic multi-player, multi-round games that we cannot literally solve (i.e., find Nash equilibria for). Can we provide any strategic guidance at all? This chapter presents a methodology for doing so. In Section 3.2 we describe the first step: identifying a space of candidate strategies. From there we describe the process of generating, through simulation, an empirical estimate of an expected payoff matrix for the reduced game in which only the generated candidate strategies are available. The process of empirically estimating the payoff matrix is very computationally expensive, and we present new and existing techniques for getting better estimates with less simulation time. These include Monte Carlo variance reduction methods and ways to avoid estimating every cell in the payoff matrix. Given an empirical estimate of a game, we discuss how we solve it using solution techniques for finite games. Finally, since the game being analyzed in this step is an empirical estimate, we present methods for assessing the confidence we can have in the solutions with respect to the underlying game we care about.

We use a First-Price Sealed-Bid Auction (FPSB) as a starting point to illustrate and provide evidence for many of the techniques in this chapter. Section 2.4 describes FPSB with two players.

[1]Compare to chess which has been considered for half a century to be game-theoretically unsolvable [Shannon, 1950]. The market games studied in this thesis have similar—in some ways more complicated—dynamic information revelation. It is in fact the lack of such dynamics in the one-shot games of Chapter 2 that allows for the analytic solutions there, despite the games' infinite action and type spaces.

In this chapter we consider the natural n-player generalization of the 2-player game defined by Equation 2.4. We now include the following theorem from the auction theory literature[2] giving the unique symmetric Nash equilibrium of FPSB. This serves as the gold standard for all of our approximation techniques.

Theorem 3.1 (McAfee and McMillan, 1987, p. 709) *The unique symmetric Nash equilibrium for FPSB with n players having types i.i.d. with cdf F, and a lowest possible type A is*

$$a(t) = t - \frac{\int_A^t F(x)^{n-1}\,dx}{F(t)^{n-1}}.$$

For $U[A,B]$ types this is $\frac{A+n-1}{n} \cdot t$, and for $U[0,1]$:[3]

$$\frac{n-1}{n} \cdot t.$$

3.1 Measuring Solution Quality: The ε Metric

Since our techniques are based on approximations to underlying games of interest, we need a way to assess solution quality. There are three broad sources of inaccuracy for our solutions: (1) we consider restricted strategy spaces and could miss better strategies (even if they are in the restricted space since we may incompletely explore it), (2) our simulation-based payoff estimates can be riddled with sampling error (especially when the simulations take minutes per game), and (3) we introduce other approximations, such as reducing the number of players (Section 3.5). For these reasons, the equilibria we derive will often not correspond to equilibria in the underlying game. In light of this, we need a measure of the quality of the strategies we find when they are not in equilibrium.

Suppose that, using some approximation $\hat{\Gamma}$ to a game Γ, we find an equilibrium profile $\hat{s}^*$ of $\hat{\Gamma}$. We want to determine how closely $\hat{s}^*$ approximates an equilibrium in Γ. If we could find an equilibrium s^* of Γ we could compare s^* and $\hat{s}^*$ directly. Of course, if we could do that then there was no need to approximate Γ in the first place. Also, the comparison between s^* and $\hat{s}^*$ is not necessarily meaningful if there is more than one equilibrium in Γ.

A more natural way to answer the question is to measure the potential gain to deviating from $\hat{s}^*$ in Γ. Ideally the answer is zero and $\hat{s}^*$ is an equilibrium for Γ as well as $\hat{\Gamma}$. In that case, $\hat{\Gamma}$ is a perfect substitute for Γ for the purpose of finding an equilibrium profile. More generally, the smaller the gain from deviating from $\hat{s}^*$ in Γ, the more faithful an approximation is $\hat{\Gamma}$. To this end, we define the ε of a profile, following Turocy's [2001] usage.

Definition 3.2 (Epsilon of a Profile) *For a symmetric game $\Gamma = \langle n, S, u() \rangle$ and strategy profile s (typically an equilibrium in some approximation of Γ), we denote by $\varepsilon_\Gamma(s)$ the potential gain to*

[2] Krishna [2002] provides a textbook treatment.

[3] *Kagel and Levin [1993] generalize the result for the $U[0,1]$ case to mth-price, single-good, sealed-bid auctions:*

$$\frac{n-1}{n+1-m} \cdot t, \tag{3.1}$$

yielding the familiar truthful bidding result for the first and second price case [Vickrey, 1961].

deviating from s in Γ:

$$\varepsilon_\Gamma(s) \equiv \max_{s' \in S} u(s', s) - u(s, s).$$

Alternatively, we can express ε as a percentage:

$$\varepsilon\%_\Gamma(s) \equiv \frac{\varepsilon_\Gamma(s)}{u(s, s)} \cdot 100.$$

This definition follows the standard notion of approximate equilibrium. Profile s is an $\varepsilon_\Gamma(s)$-Nash equilibrium of Γ, and $\varepsilon_\Gamma(s) = 0$ if and only if s is a Nash (i.e., a 0-Nash) equilibrium of Γ. Henceforth, we drop the game subscript when understood in context.

A propitious feature of the ε metric is that $\varepsilon_\Gamma(s)$ can be evaluated even when we know only a vanishing fraction of the payoff function of Γ. In particular, we only need to know the payoffs for the unilateral deviations from s. In a symmetric n-player game with S strategies, this means evaluating fewer than nS cells of the payoff matrix out of $\binom{n+S-1}{n}$, or S^n in the case of a nonsymmetric game.

3.2 Restricting the Strategy Space

A strategy is a mapping from an agent's private information (type), and the information revealed during the game, to an action. In the one-shot games considered in Chapter 2 there is no information revelation (by definition of a one-shot game) and the private information and actions are one-dimensional (real numbers). So strategies in that context are one-dimensional functions from type to action.

The first step in our methodology for empirically analyzing games is to restrict the game by constructing a finite set of candidate strategies. In FPSB, the strategy space is the set of functions from the reals to the reals (type to action). One restriction of this strategy space is to consider only strategies of the form $a(t) = kt$ for specific values of k.[4] This is a natural parameterization for FPSB, generalizing truthful bidding ($a(t) = t$) with a bid shading factor, k. To restrict the strategy space less severely, we could add a translational parameter b to allow affine linear strategies ($kt + b$). In fact, by introducing $3K - 1$ parameters, we can allow piecewise linear strategies of up to K pieces per Equation 2.2. For one-shot games with one-dimensional types (e.g., the games in Chapter 2 and their generalizations to more than two players) such piecewise linear strategy parameterizations will tend to suffice. Such parameterizations are appealing because they entail no domain knowledge—the parameters can be chosen automatically. For more complicated games, the strategy space will be too large to parameterize so naively and we rely on the manual construction of a strategy parameterization. This may be done by starting with a baseline strategy and generalizing it via parameters. The baseline strategy may be utterly simplistic, naive, and perform miserably. But as long as the space of strategies allowed by the parameterization includes smarter strategies (and they need not be identifiable as such a priori) then our methodology has hope of finding them. For FPSB, the baseline strategy of truthful bidding is as bad as not participating, guaranteeing zero utility. But introducing a simple shading parameter (without knowing a good setting for it) allows our methodology to approximate the unique symmetric Nash equilibrium of the underlying game.[5] The next section shows how.

[4] Selten and Buchta [1994] dub this *ray bidding*.

[5] There are other asymmetric equilibria [Milgrom and Weber, 1985].

In Chapter 4 we apply this method of generating candidate strategies to the Simultaneous Ascending Auctions domain, and in Chapter 6 we apply it to the Trading Agent Competition travel-shopping domain. Following is the definition of the restricted form of FPSB that we use throughout this chapter.

Definition 3.3 (FPSBn) *In the n-player first-price sealed-bid auction, player i has a private value t_i, decides to bid a_i, and obtains payoff $t_i - a_i$ if its bid is highest and zero otherwise.*[6] *FPSBn is the special case where t_i are i.i.d. $U[0,1]$, and agents are restricted to parameterized strategies, bidding $a_i(t_i) = k_i t_i$ for $k_i \in [0,1]$, for all $i \in \{1,\ldots,n\}$. We denote strategies in FPSBn by the parameter k.*

We now establish theoretical results to which we can compare the empirical methods in the rest of this chapter.

Theorem 3.4 *The expected payoff for an agent playing k_i against everyone else playing k in FPSBn is:*

$$u_i(k_i,k) = \begin{cases} \dfrac{1}{2n} & \textit{if } k_i = k = 0 \\[2ex] \dfrac{1-k_i}{n+1}\left(\dfrac{k_i}{k}\right)^{n-1} & \textit{if } k_i \leq k \\[2ex] \dfrac{(1-k_i)((n-1)k_i^2 - (n-1)k^2)}{2(n+1)k_i^2} & \textit{otherwise.} \end{cases}$$

Appendix A.5 contains the proof.

Theorem 3.5 *The best response to everyone else playing k in FPSBn is:*

$$\textit{BR}(k) \equiv \arg\max_{k_i} u_i(k_i,k) = \begin{cases} \textit{undefined} & \textit{if } k = 0 \\[2ex] \xi & \textit{if } k < \dfrac{n-1}{n} \\[2ex] \dfrac{n-1}{n} & \textit{if } k \geq \dfrac{n-1}{n}, \end{cases} \tag{3.3}$$

where $\xi \equiv$

$$\frac{\sqrt[3]{3}\left(k^2\left(n^2-1\right)\left(9n+\sqrt{3(n+1)\left((n-1)k^2+27(n+1)\right)}+9\right)\right)^{2/3} - 3^{2/3}k^2\left(n^2-1\right)}{3(n+1)\sqrt[3]{k^2(n-1)\left(9n^2+18n+(n+1)^{3/2}\sqrt{3(n-1)k^2+81(n+1)}+9\right)}}.$$

[6] *In the case of ties (though these happen with probability zero in the infinite game with nondegenerate strategies), one of the bidders with the highest bid is chosen with equal probability (and thus in expectation each high bidder i receives $(t_i - a_i)/k$ in a k-way tie). Formally, the full payoff function is:*

$$u_i(\boldsymbol{t},\boldsymbol{a}) = \begin{cases} \dfrac{t_i - a_i}{|\{x \in \boldsymbol{a} \mid x = \max(\boldsymbol{a})\}|} & \textit{if } a_i = max(\boldsymbol{a}) \\[2ex] 0 & \textit{otherwise.} \end{cases} \tag{3.2}$$

Appendix A.6 contains the proof. For the special case of $n = 3$, Selten and Buchta [1994] point out the third case of (3.3), that the best response to playing k greater than the equilibrium of $2/3$ is to play $2/3$. Turocy [2001] generalizes this result to arbitrary n and for mth-price, single-good, sealed-bid auctions: when all other agents are playing k greater than the equilibrium (see Equation 3.1) the best response is to bid the equilibrium k. This result yields the third case of (3.3) as a special case ($m = 1$). Additionally, Turocy derives the second case, ξ, for the special case $n = 2$, albeit with a typo. The following is the best response to $k \in (0, \frac{n-1}{n})$ for $n = 2$:

$$\xi = \frac{k^2}{3\sqrt[3]{\sqrt{k^6 + 81k^4} - 9k^2}} - \frac{1}{3}\sqrt[3]{\sqrt{k^6 + 81k^4} - 9k^2}.$$

Turocy [2001, p. 107] also gives the plot of (3.3) for $n = 2$.

Interestingly, the unrestricted best response, i.e., the best response to k in the unrestricted FPSB, is conceptually much simpler despite being nonlinear: $a(t) = \min(\frac{n-1}{n}t, k)$ for $k > 0$. (We found this using the best-response algorithm in Chapter 2 for the 2-player case, and inferred/verified it via Monte Carlo estimation (Appendix B) for more than two players.)

Corollary 3.6 *The ε of a symmetric profile in* FPSBn, $\varepsilon_{\text{FPSB}n}(k)$, *is:*

$$\max_x(u_i(x,k)) - u_i(k,k) = \begin{cases} \frac{1}{2} - \frac{1}{2n} & \text{if } k = 0 \\ \frac{(k-\xi)\left(\xi^2 - k\xi + \xi + k + (\xi-1)(\xi+k)n\right)}{2\xi^2(n+1)} & \text{if } k < \frac{n-1}{n} \\ \frac{1 - n + k\left(\left(\frac{n-1}{kn}\right)^n + n - 1\right)}{n^2 - 1} & \text{otherwise.} \end{cases}$$

Proof. Having closed-form solutions for $u_i(k_i, k)$ and $\text{BR}(k)$—Theorems 3.4 and 3.5—a closed-form solution for ε follows almost immediately. Since $u_i(x, 0)$ has no maximum, we take the supremum to be implied, i.e., $\lim_{x\to 0} u_i(x, 0) = E[T_i] = 1/2$. Otherwise, $\max_x(u_i(x,k)) = u_i(\text{BR}(k), k)$. □

The following lemma allows us to establish the unique symmetric equilibrium of FPSBn and is also needed for the theorems in Section 3.5.

Lemma 3.7 *Let f be the function in the second case of* $\varepsilon_{\text{FPSB}n}(k)$,

$$f(k) \equiv \frac{(k-\xi)\left(\xi^2 - k\xi + \xi + k + (\xi-1)(\xi+k)n\right)}{2\xi^2(n+1)}.$$

Then $f\left(\frac{n-1}{n}\right) = 0$ *and f has no other positive roots.*

Appendix A.7 contains the proof. That $\frac{n-1}{n}$ is a symmetric equilibrium of FPSBn is already implied by the fact that it is an equilibrium in unrestricted FPSB with $U[0, 1]$ types (Theorem 3.1) since FPSBn is the same game as FPSB but with a subset of the actions). But this does not establish uniqueness.

Theorem 3.8 $\frac{n-1}{n}$ *is the unique symmetric equilibrium of* FPSBn.

Proof. The theorem is trivially true for $n = 1$. Consider the $n > 1$ case. That $\frac{n-1}{n}$ is an equilibrium follows directly from Theorem 3.5 (alternatively, from Theorem 3.1). For uniqueness it suffices to show that for no other k is BR$(k) = k$. Theorem 3.5 rules out $k = 0$ and Lemma 3.7 establishes that $\varepsilon(k) \neq 0$ for $k \in (0, \frac{n-1}{n})$. □

3.3 Game Simulators and Brute-Force Estimation of Games

So far, we have taken a baseline strategy (truthful bidding) for FPSB and generalized it with a natural parameterization (bid shading fraction). We solve this restricted form of FPSB in the previous section, showing that the restriction does not change the unique symmetric equilibrium. Still, FPSBn is an infinite game so we further restrict the strategy space by discretizing the parameters. We now show how to approximate a normal form version of the game via simulation. The closed-form expressions in the previous section allow us to compare with the limiting case of infinite simulation time and continuous parameters.

The first step in estimating the empirical game is to write a game simulator.[7] A game simulator is a function that takes as input the agent types and the agent strategies and outputs payoffs. For example, a game simulator for FPSBn can be written as a straightforward extension to the payoff function in Definition 3.3 as follows:[8]

$$\text{FPSB}n(\boldsymbol{t}, \boldsymbol{k})_i = \begin{cases} t_i - k_i t_i & \text{if } k_i t_i = max(\{k_1 t_1, \ldots, k_n t_n\}) \\ 0 & \text{otherwise,} \end{cases}$$

where k_i is the shade parameter employed by agent i. In this case, the implementation of the simulator is trivial and hardly warrants the name. In subsequent chapters we consider cases where the payoff functions involve complicated interactions between multiple auctions and the strategies alone run to hundreds or thousands of lines of code. Nonetheless, the principle remains the same: the simulator determines agent actions by applying the strategies to the agent types (and information dynamically revealed during the game) and from the actions determines outcomes, i.e., payoffs.

In order to exploit some of the variance reduction techniques in the next section, we additionally require that the game simulator be deterministic. This is without loss of generality as we can derive any desired randomness in the game from the dummy player, Nature, which is given the appropriate type distribution (Section 1.1). Framed as such, symmetry is broken, but this is overcome simply by counting Nature only as a player for the purposes of type assignments but not in the list of strategies.

The most naive, brute-force approach to empirical game estimation has the following steps:

1. For each strategy profile (distinct combination of candidate strategies), repeat for a large number of samples:

    a) Generate random preferences for each agent according to the type distribution (which is part of the game definition).

[7] This may be thought of as the first step in the mapping of a Bayesian game to a normal form game (Definition 1.1).
[8] With the same adjustment for tie breaking as in Equation 3.2.

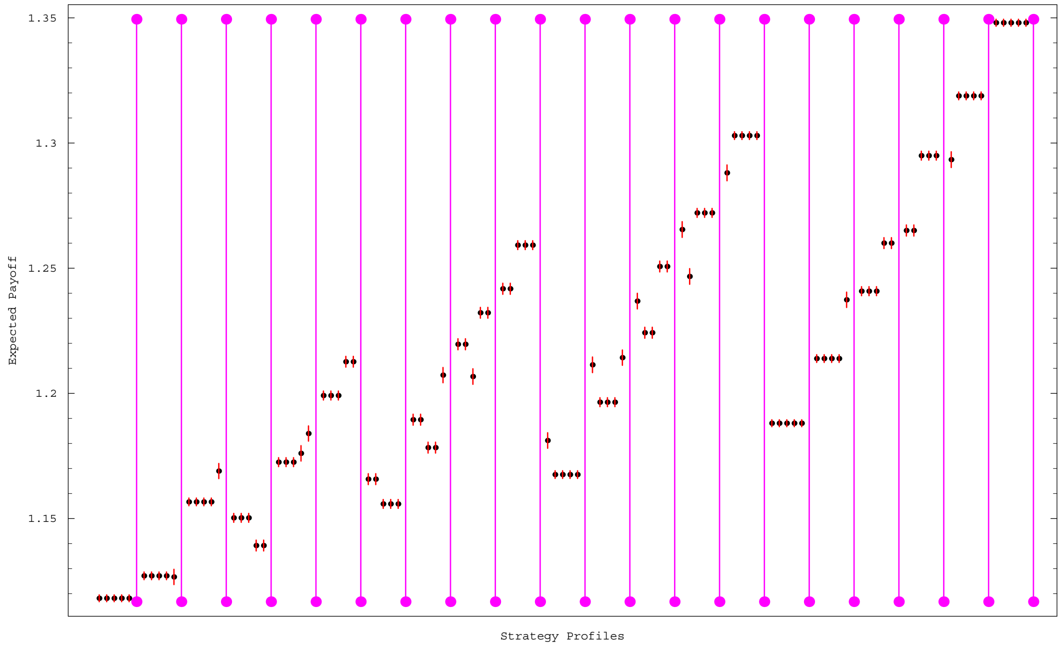

Figure 3.1: Payoff matrix for an SAA game with 5 players and 3 strategies. Each column corresponds to a strategy profile with the symmetric profile of all playing strategy 1 on the left and the profile with all 3 on the right. The jth dot within a column represents the mean payoff for the jth strategy in the profile. This payoff matrix is based on over 45 million games simulated for each of the 21 profiles, requiring weeks of cpu time. The error bars denote 95% confidence intervals. For this case, the unique Nash equilibrium (all playing strategy 3) is apparent by inspection.

b) If applicable, generate the other random elements of the game (i.e., Nature's type).

c) Run the game simulator to produce sample payoffs.

2. Average the sample payoffs for each profile.

In other words, use brute-force (also called crude or plain) Monte Carlo sampling to estimate the expected payoffs in each cell of the payoff matrix. The above procedure yields an empirical estimate of the restricted game. Figure 3.1 shows an example of an empirical payoff matrix for an SAA game (cf. Chapter 5). Note that as the set of candidate strategies approaches the full set of allowed strategies in the underlying game, the empirical payoff matrix approaches the full strategic (normal) form representation. Furthermore, Armantier *et al.* [2000, to appear] show that a Nash equilibrium in the restricted game (what they call a Constrained Strategic Equilibrium) approaches an equilibrium in the underlying game as the allowed strategy set approaches the unrestricted set.

As an example of the applicability of step 1b, suppose that coin flips are used to break ties in FPSBn. We could cast this as a game with Nature having discrete type chosen uniformly from the $n!$ permutations of $\{1, \ldots, n\}$. In the case of ties, the player appearing first in Nature's list wins. For simple games like FPSB and others in Chapter 2, we can typically keep the utility function deterministic by assuming that the agents split the surplus in the case of a tie, this being equivalent in expectation to deciding the winner randomly. For more complicated games such as the Trading Agent Competition (see Chapter 4), the random elements are not so easily distilled out.[9]

For easier applicability of the variance reduction techniques in the next section, we restrict Nature's type to be a single $U[0,1]$ random variable. A simple trick makes this w.l.o.g. for arbitrary tie-breaking or other random elements in a game. Namely, the mechanism includes a pseudorandom number generator (RNG) that is seeded with Nature's type. Then, for example, in the case of ties the mechanism assigns its next pseudorandom (i.e., not random) number to each of the tying bidders in turn and then picks the one with the highest number. Including a RNG in the game simulator is a matter of adding a single equation. We propose `ran0` from Numerical Recipes [Press *et al.*, 1992]:

- Constants: $a = 7^5$, $m = 2^{31} - 1$.
- Initialize r to Nature's type times m.
- Successive random numbers: $r \leftarrow a \cdot r \pmod{m}$.

This simple RNG is considered simulation-quality as long as the 2^{31} ($\sim$ 2 billion) period suffices, which it will for any reasonable game—certainly all the games, including TAC, considered in this thesis. Specifically, we require that a single instance of the game uses fewer than 2^{31} random numbers. (If not, a more sophisticated RNG would be employed [Press *et al.*, 1992].)

We test the brute-force estimation method on FPSB4 by restricting the game to 41 strategies ($k \in \{0, 1/40, \ldots, 1\}$) simulating the game 15,000 times for each of the $\binom{4+41-1}{4} = 135,751$ strategy profiles. Figure 3.2 shows $\varepsilon(k)$ for every pure symmetric strategy profile—all agents playing k (i.e., bidding $k \cdot t$). This method indeed converges on the equilibrium of the underlying game, but the computational burden is significant, even with such a simple game as FPSB.

3.4 Variance Reduction in Monte Carlo Sampling

The core problem in game estimation is determining the payoffs in the cells of the payoff matrix—that is, estimating the expected payoffs for a fixed strategy profile given the distribution of agents' and Nature's types. Given that we have defined a function—the game simulator—that performs such a mapping, there are a number of off-the-shelf techniques that can be applied [Ross, 2001; Press *et al.*, 1992; Galassi *et al.*, 2002; L'Ecuyer, 1994]. These are known as variance reduction techniques for Monte Carlo estimation. They serve to improve the efficiency of the procedure presented in Section 3.3, potentially reducing by orders of magnitude the number of samples (game simulations) to estimate a payoff matrix at a given level of statistical significance.

Control Variates

The method of control variates is one such standard variance reduction technique [Ross, 2001]. It improves the estimate of expected payoffs by exploiting correlation of payoffs with summaries of the randomness in the game. The idea is to replace sampled payoffs with payoffs that have been *adjusted for "luck"* such that the mean payoffs are unaffected but the variance is reduced. For example, in FPSB, we expect an agent's valuation to correlate positively with its surplus. Thus, we bump up an agent's payoff when it has a low valuation and scale it down when it has a high

[9] In fact, this would be quite hopeless—though possible in principle, just as in principle it is possible to represent the full TAC game or chess in strategic form.

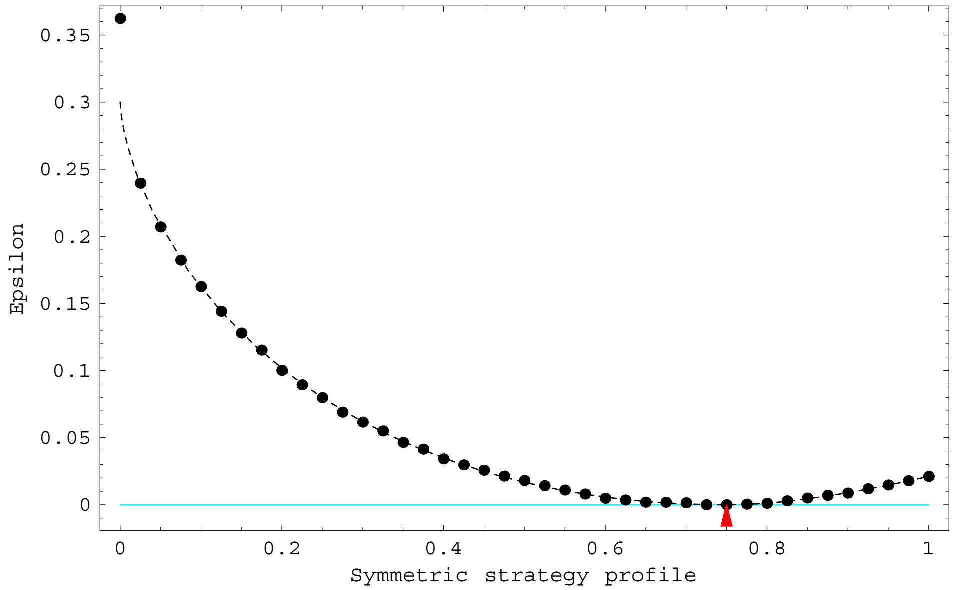

Figure 3.2: Evaluation of the pure symmetric (homogeneous) profiles for an empirical estimation of FPSB4. The arrow at $k = 3/4$ shows the unique symmetric equilibrium for the underlying infinite game.

valuation such that the positive and negative adjustments average out to zero or near zero. This straightforwardly reduces variance in the payoffs. By sampling these adjusted payoffs it will tend to take fewer samples to converge to a good approximation of the true expected payoffs.

Consider a game Γ defined by a game simulator $\Gamma(\boldsymbol{t}, s)$ mapping types (agents plus Nature) and strategy profiles to payoffs. In general, we seek a function g from types to payoff *estimates*. For every strategy profile s, we then use g to adjust each sampled payoff $\Gamma(\boldsymbol{t}, s)$ by subtracting $g(\boldsymbol{t}) - E_{\boldsymbol{t}}[g(\boldsymbol{t})]$. If the game simulator itself were used as g, i.e., $g(\boldsymbol{t}) \equiv \Gamma(\boldsymbol{t}, s)$, then every sampled payoff would get replaced by the true expected payoff. Of course, if we knew the expectation of the game simulator we would not need Monte Carlo approaches in the first place. So the trick is to find a g for which we can derive $E[g(\boldsymbol{t})]$, or at least estimate it more easily and accurately than we can $E[\Gamma(\boldsymbol{t}, s)]$. Theorem 3.4 provides a closed-form expression for $E[\text{FPSB}n(\boldsymbol{t}, s)]$ for symmetric profiles s and unilateral deviations from symmetric profiles. So for FPSBn we really do know the expectation of the game simulator. That gives us the full empirical payoff matrix to infinite precision and for arbitrarily fine discretization of the strategy space.

Even for such a simple game as FPSBn, a closed-form solution for expected payoff was not trivial to find. Next best, suppose we could determine analytically how expected payoff varies with a single agent's type. In FPSB2 the expected payoff for an agent with specific type t playing k against an opponent with type $T \sim U[0, 1]$ playing k' is $(t - kt)Pr(kt > k'T) = (1 - k)kt^2/k'$. (Note the difference from Theorem 3.4 in that we now consider expected payoff for a specific type rather than finding the expectation over own type.) Letting

$$g(t) = \frac{(1 - k)kt^2}{k'},$$

we integrate to find

$$E[g(t)] = \frac{(1-k)k}{3k'}.$$

This derivation generalizes straightforwardly to n players (from the perspective of player i):

$$g(t_i) = \frac{(1-k_i)k_i^{n-1}t_i^n}{\prod_{j \neq i} k_j}, \tag{3.4}$$

$$E[g(t_i)] = \frac{(1-k_i)k_i^{n-1}}{(n+1)\prod_{j \neq i} k_j}. \tag{3.5}$$

(There are 3 special cases: (1) $g(t_i) = 0$ if $k_i = 0$ and for some $j \neq i$, $k_j > 0$, (2) $g(t_i) = \frac{1}{2n}$ if $k_i = 0 \ \forall i \in \{1, \ldots, n\}$, (3) otherwise, with z players playing $k = 0$, ignore them and substitute $n \leftarrow n - z$ above.)

For more complex games (such as those addressed in subsequent chapters), we would not typically be able to derive any estimator g analytically. Instead we can estimate it with an initial run of brute-force Monte Carlo samples and fit a function to the data. This is more likely to be feasible if g does not depend on the strategy profile so that we do not need to generate a distinct data set (pairings of types and payoffs) for every profile. Of course, payoff does depend on the strategies, but since g needs only to estimate the payoffs we can typically find a function from types alone that correlates with payoff. To avoid introducing bias in the adjusted payoffs we use a distinct set of samples to estimate g, but even if we use the same set of samples the bias goes to zero as the number of samples increases [L'Ecuyer, 1994]. Returning to FPSB, we might generate a data set pairing a single agent's type with payoff across all or a random subset of profiles and perform simple linear regression to find $g(t) = \beta t + \alpha$. This means that a payoff x is adjusted to $x - (g(t) - E[g(t)]) = x - \beta t - \alpha + \beta E[t] + \alpha = x - \beta(t - E[t])$. In general, any constant term in g cancels out and so from now on we ignore them. In particular, we use $g(t) = \beta t$ instead of $\beta t + \alpha$.

Preliminary experiments with FPSB4 (and strategy granularity of $1/20$) support the hypothesis that the better g approximates the true expected payoff function, the greater the variance reduction. For example, for the equilibrium profile ($k = 3/4$), linear regression (the control variate method requiring no domain knowledge, analytic results, or extensive simulation) achieved a 41% reduction in variance. Linear regression on samples exclusively from the equilibrium profile happened to yield a similar enough control variate that it achieved negligible additional variance reduction. The linear models achieve most of the gain possible even by the analytically determined control variate of Equations 3.4 and 3.5 which achieves a 56% reduction in variance. Figure 3.3 plots the control variates:

1. $g(t) = 0$, i.e., no control variate adjustment.
2. $g(t) = 0.17t - 0.03$, determined by linear regression indiscriminate of profile.
3. $g(t) = 0.20t - 0.05$, determined by linear regression for samples exclusively from the equilibrium profile (all playing $k = 3/4$).
4. $g(t) = t^4/4$, which is Equation 3.4 with $n = 4$ and $\boldsymbol{k} = \langle \frac{3}{4}, \frac{3}{4}, \frac{3}{4}, \frac{3}{4} \rangle$.

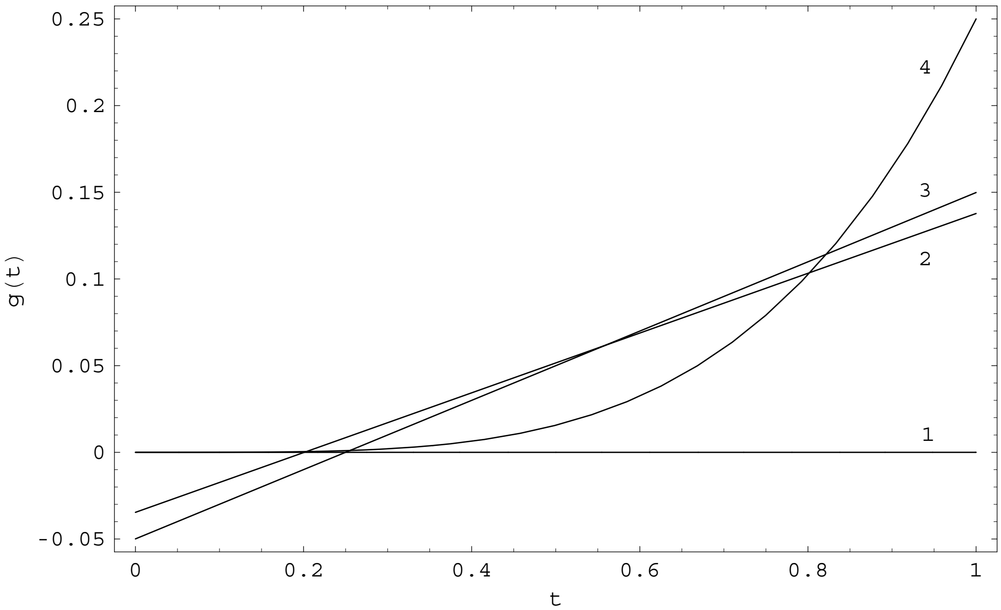

Figure 3.3: Four control variates, $g(t)$, for FPSB4.

And Figure 3.4 shows the variances of adjusted payoff for the above control variates along with (as a sanity check)

5. $g(t) = \text{FPSB4}(\boldsymbol{t}, \langle \frac{3}{4}, \frac{3}{4}, \frac{3}{4}, \frac{3}{4} \rangle)$, for which Theorem 3.4 gives $E[g(t)]$.

In general, with multidimensional types, including Nature's, it may not be obvious how these random elements influence payoffs. The dimensionality does not have to be very high before it becomes very hard to empirically determine meaningful relationships between types and payoffs. For example, imagine a game involving bidding for many goods with an agent's type being the vector of valuations for each. Depending on the specifics of the game we might expect the sum or the max of an agent's valuations to correlate with payoff. It would take a sophisticated learning algorithm with a lot of data (i.e., many game simulations) to rival a simple linear regression from sum or max valuation to payoff. Thus, when we have sufficient domain knowledge—such as knowing that the sum or the max are good summary statistics—we introduce control variates manually. In Chapter 6 we apply the method of control variates to the Trading Agent Competition game.

Other Methods: Quasi-Random and Importance Sampling

Additional variance reduction methods that we describe here but do not currently employ in any experiments are quasi-random sampling and importance sampling [L'Ecuyer, 1994]. For the following variance reduction methods, we assume that the type space is a vector of $U[0, 1]$ random variables. This is without loss of generality since for any random variable X with cdf F, the distribution of $F^{-1}(Y)$ where $Y \sim U[0, 1]$ is identical to that of X [Rice, 1995].[10] We then simply fold F^{-1} into

[10] Assuming that there is an interval I (not necessarily finite) such that F is zero to the left of I, one to the right, and strictly increasing on I.

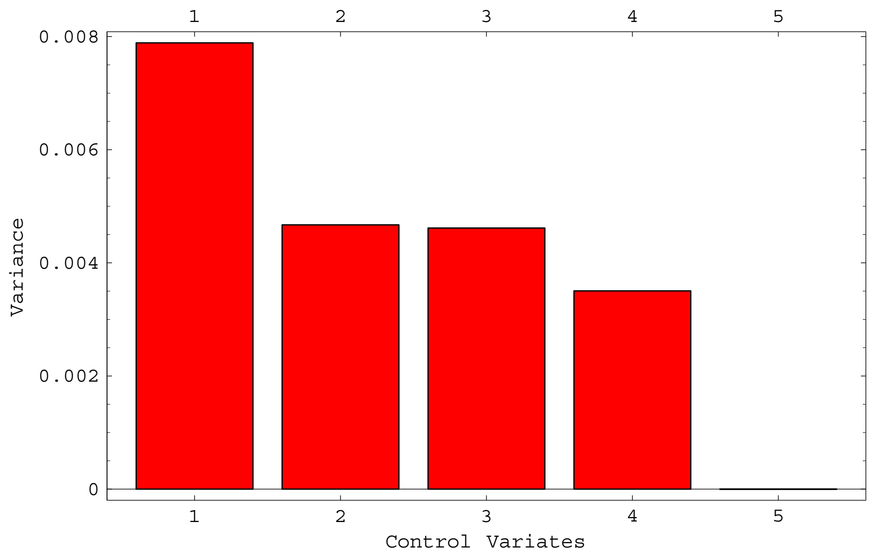

Figure 3.4: Empirically determined variances for five control variates for the equilibrium profile, $\boldsymbol{k} = \langle \frac{3}{4}, \frac{3}{4}, \frac{3}{4}, \frac{3}{4} \rangle$, in FPSB4 with strategies $k \in \{0, \frac{1}{20}, \ldots, 1\}$. Methods 1 and 2 do not require analytic results nor prohibitive sampling. Method 3 requires no analytic results but requires many samples at every profile to which it is applied. Method 4 requires analytic results that are only attainable for simple games. When Method 5 is feasible, no simulation is needed at all.

the definition of the game simulator. After this transformation, the domain of the game simulator is an n-dimensional unit hypercube of $U[0, 1]$ random variables and thus finding the expectation is equivalent to finding the n-tuple integral of the game simulator.

Quasi-random sampling means sampling the types along a uniform hypergrid rather than according to the uniform distributions. Grid sampling requires an upfront decision on granularity; however, quasi-random sequences such as Sobel sequences avoid this by generating iteratively finer grids, each iteration filling in the gaps from the previous iteration. The benefit (and the source of the variance reduction) is that quasi-random sequences fill a Euclidean space more uniformly than uncorrelated random points [Press *et al.*, 1992].

Importance sampling means sampling according to a carefully constructed alternate distribution and then applying an adjustment to return to the original distribution. The principle is that the expectation of a random variable P with pdf p is equal to the expectation of $Xp(X)/f(X)$ where X is any other random variable—in particular the carefully constructed one—with pdf f. The alternate distribution is constructed to concentrate sampling effort in regions of the domain of the function where more samples improve the expected-value estimate most quickly. This is estimated using initial brute-force Monte Carlo sampling.

3.5 Reducing the Number of Players

In Section 3.2 we reduce the strategy space to tame the complexity of very large games. Here we present a related technique: reducing the number of players. The idea is that although a strategy's payoff depends on the play of other agents (otherwise we are not in a game situation at all), it may be relatively insensitive to the exact numbers of other agents playing particular strategies. For example, let $(s, k; s')$ denote a profile where k other agents play strategy s, and the rest play s'. In many natural games, the payoff for playing any particular strategy against this profile will vary smoothly with k, with only incremental differences between contexts $(s, k-1; s')$, $(s, k; s')$, and $(s, k+1; s')$. If such is the case, we sacrifice relatively little fidelity by restricting attention to subsets of profiles, for instance those with only even numbers of any particular strategy. To do so essentially transforms the n-player game to an $n/2$-player game over the same strategy set, where the payoffs to a profile in the reduced game are simply those from the original game where each strategy in the reduced profile is played twice.

The potential savings from analyzing reduced games are considerable, as they contain combinatorially fewer profiles. In Chapter 6 we apply hierarchical player reduction to the TAC travel-shopping domain where it makes all the difference in being able to analyze the game. In the following sections, we define player reduction formally and present evidence of the viability of approximating games by reduced versions.

Player Reduction Definition and Hierarchical Reduction

We develop the concept of player reduction in the framework of symmetric normal-form games[11] (Definition 1.2) and define a reduced game as follows.

Definition 3.9 (Player-Reduced Game) *Let $\Gamma = \langle n, S, u() \rangle$ be an n-player symmetric game, with $n = pq$ for integers p and q. The p-*player reduced version of Γ*, written $\Gamma\!\downarrow_p$, is given by $\langle p, S, \hat{u}() \rangle$, where*

$$\hat{u}_i(s_1, \ldots, s_p) = u_{q \cdot i}(\underbrace{s_1, \ldots,}_{q} \underbrace{s_2, \ldots,}_{q} \ldots, \underbrace{s_p, \ldots}_{q}).$$

In other words, the payoff function in the reduced game is obtained by playing the specified profile in the original q times.

The idea of a reduced game is to coarsen the profile space by restricting the degrees of strategic freedom. Although the original set of strategies remains available, the number of agents playing any strategy must be a multiple of q. Every profile in the reduced game is one in the original game, of course, and any profile in the original game can be reached from a profile contained in the reduced game by changing at most $p(q-1)$ agent strategies.

We can apply player reduction iteratively and get a hierarchy of reduced games. We note that the game resulting from a series of reductions is independent of the reduction ordering. Let $q = r \cdot r'$. Then

$$(\Gamma\!\downarrow_{p \cdot r})\!\downarrow_p = (\Gamma\!\downarrow_{p \cdot r'})\!\downarrow_p = \Gamma\!\downarrow_p .$$

Consider FPSBn with $n = pq$. In the reduced game FPSB$n\!\downarrow_p$, each agent $i = 1, \ldots, p$ selects a single action k_i, which then gets applied to q valuations $v_{i_1}, \ldots, v_{i_q}$ to define q bids. The auction proceeds as normal, and agent i's payoff is defined as the *average* payoff associated with its q bids. Note that the game FPSB$n\!\downarrow_p$ is quite a different game from either FPSBn or FPSBp. When represented explicitly over a discrete set of actions, FPSB$n\!\downarrow_p$ is the same size as FPSBp, and both are exponentially smaller than FPSBn.

---

[11] Although the methods may generalize to some degree to partially symmetric games, or to allow partial reductions in games given in extensive form, we do not pursue such extensions here.

Theoretical and Experimental Evidence

The premise of our approach is that the reduced game will often serve as a good approximation of the full game it abstracts. We know that in the worst case it does not. In general, an equilibrium of the reduced game may be arbitrarily far from equilibrium with respect to the full game, and an equilibrium of the full game may not have any near neighbors in the reduced game that are close to equilibrium there.[12] The question, then, is whether useful hierarchical structure is present in "typical" or "natural" games, however we might identify such a class of games of interest.

FPSB

We begin by examining $\text{FPSB}n$ which has a unique symmetric Nash equilibrium of $k = \frac{n-1}{n}$ (Theorem 3.8). For example, the equilibrium for $\text{FPSB}2$ is $1/2$, and for $\text{FPSB}4$ it is $3/4$. From the following theorem, giving the equilibrium of $\text{FPSB}n{\downarrow_p}$, we have $2/3$ in equilibrium for $\text{FPSB}4{\downarrow_2}$.

Theorem 3.10 *The unique symmetric Nash equilibrium of* $\text{FPSB}n{\downarrow_p}$ *is*

$$\frac{n(p-1)}{p+n(p-1)}.$$

Appendix A.8 contains the proof. Note the special cases that the equilibrium of $\text{FPSB}n{\downarrow_1}$ is $k = 0$ and the equilibrium of $\text{FPSB}n{\downarrow_n}$, i.e. $\text{FPSB}n$, is $k = \frac{n-1}{n}$. (Thus Theorem 3.10 generalizes Theorem 3.8 which gives the Nash equilibrium of $\text{FPSB}n$.)

The difference between equilibrium strategies of $\text{FPSB}4{\downarrow_2}$ and $\text{FPSB}4$ is one measure of their distance. Returning to our ε metric (Definition 3.2), Figure 3.5 plots $\varepsilon(k)$ for the three game variations. Theorem 3.6 gives us closed-form expressions for $\varepsilon_{\text{FPSB}n}(k)$, whereas the curve for $\varepsilon_{\text{FPSB}4{\downarrow_2}}$ was estimated by applying the methods of Sections 3.2 and 3.3 (though we know its root exactly by Theorem 3.10). We generated candidate strategies by discretizing k at intervals of $1/40$. At this granularity, $\text{FPSB}4$ comprises 158 times as many profiles as does $\text{FPSB}4{\downarrow_2}$. We estimated the payoff function by sampling 36,000 games per profile (5 billion total) using brute-force Monte Carlo. This was sufficient to estimate the unique symmetric equilibrium of $\text{FPSB}4{\downarrow_2}$ as being between 0.625 and 0.675 (the exact value being $2/3$). Based on our analysis, $\text{FPSB}4{\downarrow_2}$ compares quite favorably to $\text{FPSB}2$ as an approximation of $\text{FPSB}4$. In particular, taking their respective equilibrium values, $\varepsilon_{\text{FPSB}4}(2/3)$ is nearly ten times smaller than $\varepsilon_{\text{FPSB}4}(1/2)$.

We now generalize this conclusion—that the 2-player reduction better approximates the 4-player game than does the 2-player game—to arbitrary n and p. In the following theorems, $\varepsilon()$ refers to $\varepsilon_{\text{FPSB}n}()$ and we denote the unique symmetric equilibrium of a game Γ by $eq(\Gamma)$, defined only for symmetric games with unique symmetric equilibria.

Lemma 3.11 *The function* $\varepsilon(k)$ *is strictly decreasing for* $0 < k \leq \dfrac{n-1}{n}$ *for all* $n > 1$.

Appendix A.9 contains the proof, which relies on Lemma 3.7.

Having established this lemma, the following theorem shows that for any number of players n, a player-reduced version (Definition 3.9) of $\text{FPSB}n$ yields better strategies (both in proximity to the

[12] $\text{FPSB}n{\downarrow_1}$ is an example (albeit a degenerate one) of a reduced game having very different equilibria than the full game. The optimal strategy in the 1-player reduction is to bid zero (as it is in the actual 1-player version of FPSB) whereas in the n-player game, for $n > 1$, it is in equilibrium to bid a large fraction (at least $1/2$) of one's type.

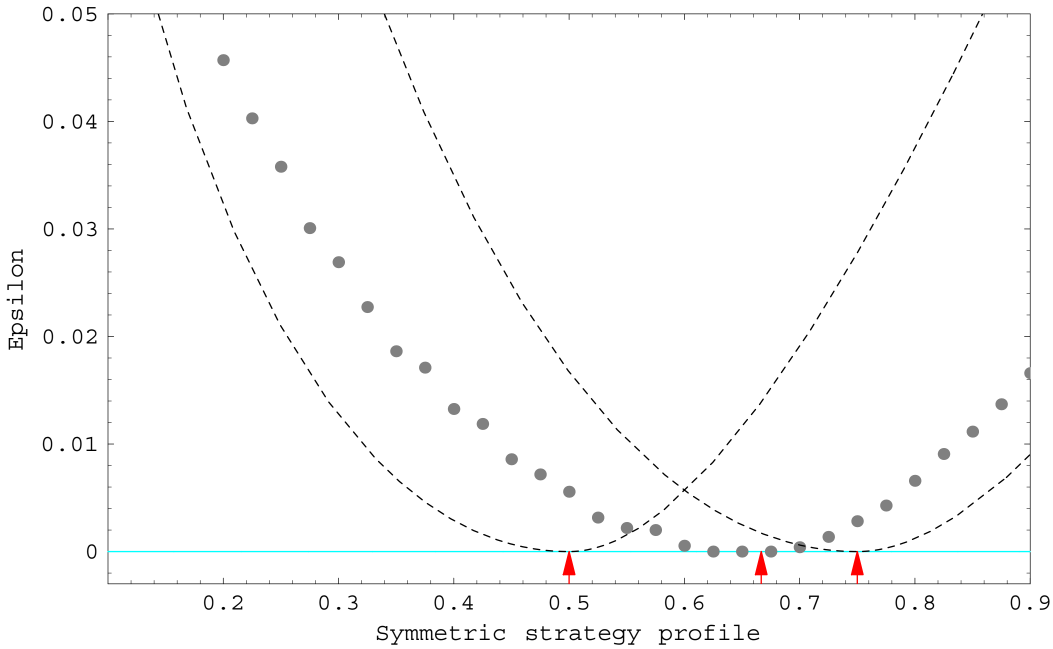

Figure 3.5: Epsilons for symmetric profiles of FPSB2 (left dashed line), FPSB$4{\downarrow_2}$ (dots), and FPSB4 (right dashed line). Unique symmetric equilibria ($1/2$, $2/3$, $3/4$) are indicated by arrows on the x-axis.

true equilibrium and in terms of lower ε) than does the version of the game with the number of players reduced outright.[13]

Theorem 3.12 *For all integers n and p with $n > p \geq 1$,*

$$eq(\text{FPSB}p) < eq(\text{FPSB}n{\downarrow_p}) < eq(\text{FPSB}n) \tag{3.6}$$

and

$$0 = \varepsilon(eq(\text{FPSB}n)) < \varepsilon(eq(\text{FPSB}n{\downarrow_p})) < \varepsilon(eq(\text{FPSB}p)). \tag{3.7}$$

Appendix A.10 contains the proof of (3.6), with (3.7) being an immediate consequence of (3.6) and Lemma 3.11. Additionally, we show that for all FPSBn, less drastic reductions are better approximations than more drastic reductions.

Theorem 3.13 *For all integers n, p, and q with $n > p > q \geq 1$:*

$$eq(\text{FPSB}n{\downarrow_q}) < eq(\text{FPSB}n{\downarrow_p}) < eq(\text{FPSB}n) \tag{3.8}$$

and

$$0 = \varepsilon(eq(\text{FPSB}n)) < \varepsilon(eq(\text{FPSB}n{\downarrow_p})) < \varepsilon(eq(\text{FPSB}n{\downarrow_q})). \tag{3.9}$$

[13] As suggested by Scott Page, we can sometimes do even better by first increasing the number of players before applying player reduction. This works for some instances of FPSB but it would be hard to justify application of this trick in games for which we don't have a gold standard (the equilibrium of the full game) to compare to.

Appendix A.11 contains the proof for the first inequality of (3.8). As in Theorem 3.12, (3.9) follows directly from (3.8) and Lemma 3.11. The second inequality of (3.8) and the first inequality of (3.9) are given by Theorem 3.12.

Local-Effect Games

The preceding analysis is reassuring, but of course we do not actually need to approximate FPSBn, since its general solution is known. To further evaluate the quality of reduced-game approximations, we turn to other natural games of potential interest. Facilitating such studies was precisely the motivation of the authors of GAMUT [Nudelman *et al.*, 2004], a flexible software tool for generating random games from a wide variety of well-defined game classes. Using GAMUT, we can obtain random instances of some class, and examine the relation of the original games to versions reduced to varying degrees. The advantage of a generator such as GAMUT is that we can obtain a full game specification quickly (unlike for TAC), of specified size based on our computational capacity for analysis. Moreover, we can sample many instances within a class, and develop a statistical profile of the properties of interest.

Here we focus on a particular class known as *local-effect games* (LEGs) [Leyton-Brown and Tennenholtz, 2003], a localized version of congestion games motivated by problems in AI and computer networks. Specifically, we consider symmetric bi-directional local-effect games randomly generated by GAMUT by creating random graph structures and random polynomial payoff functions decreasing in the number of action-nodes chosen.

In a preliminary experiment, we generated 15 symmetric LEG instances with six players and two strategies, and payoffs normalized on $[0, 1]$. For each of these we generated the corresponding 3-player reduction. We then fed all 30 of these instances to GAMBIT [McKelvey *et al.*, 1992, and see Section 2.1] which computed the complete set of Nash equilibria for each. In 11 of the original games, all equilibria are pure, and in these cases the equilibria of the reduced games match exactly. In the remaining four games, GAMBIT identified strictly mixed equilibria in both the full and reduced games. In two of these cases, for every equilibrium in the full game there exists an equilibrium of the reduced game with strategy probabilities within 0.1. In the remaining two games, there are long lists of equilibria in the full game and shorter lists in the corresponding reduced games. In these cases, most but not all of the equilibria in the reduced game are approximations to equilibria in the full game.

In broader circumstances, we should not expect to see (nor primarily be concerned with) direct correspondence of equilibria in the original and reduced games. Thus, we evaluate the approximation of a reduced game in terms of the average ε in the original game over all its equilibrium profiles in the reduced game. Note that to calculate this measure we need not be able to solve the full game. Since the games under consideration are symmetric, our assessment includes only the symmetric equilibria, where every agent plays the same (mixed) strategy.[14]

We evaluated this metric for 2-strategy local-effect games with n players, for $n \in \{4, 6, 8, 10, 12\}$. Figure 3.6 shows the result of generating between 200 and 10,000 games in each of the n-player game classes and computing the ε metric for every possible reduction of every game, starting with the most drastic reduction—to one player—and ending with the highest-fidelity reduction, i.e., to half as many players. We also include the average ε for the social optimum (the profile maximizing aggregate payoff) in each game class as calibration. We find that the social optimum fares better

[14] Symmetric games necessarily have symmetric equilibria—see Appendix C—though they may have asymmetric equilibria as well.

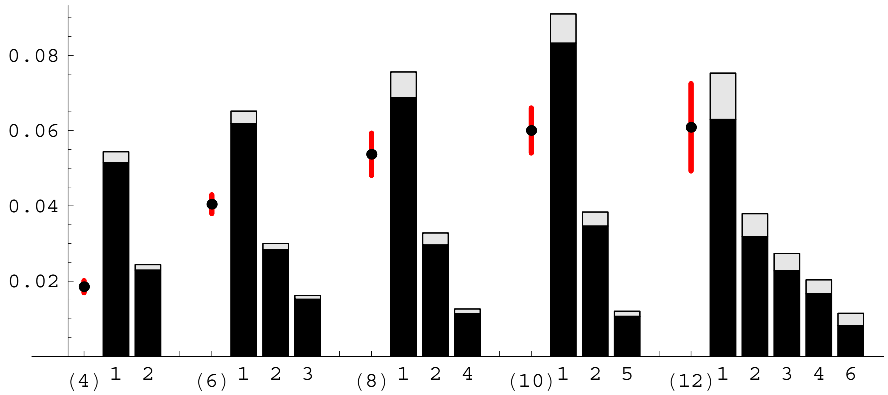

Figure 3.6: Local-effect games with 4, 6, 8, 10, and 12 players. Each group of bars shows the average ε for equilibria of reductions of the given game at increasing fidelity. The number of players in the full game is shown in parentheses, with the number in reduced games under each bar. The bars extend upward to indicate a 95% confidence upper bound on ε. To the left of each group is shown the ε (with 95% confidence interval) of the social optimum of the full game.

than the equilibria in the 1-player reduction (i.e., the strategy yielding the highest payoff if played by everyone) but that all the higher fidelity reductions yield equilibria with average ε significantly outperforming the social optima. The only exception is the case of approximating 4-player games with their 2-player reductions. We note that in fully 90% of the 4-player LEG instances, the social optimum is also an equilibrium, making it particularly hard to beat for that game class. The percentages are also high in the other classes—decreasing with the number of players to 77% for the 12-player instances—yet the social optima in all the other classes are beaten by solutions to the reduced games.

In addition to supporting the hypothesis that we can approximate a game by its reduction, we conclude from this analysis that, as in FPSBn, we get diminishing returns on refinement. There is a large benefit to going from 1- to 2-player games (i.e., bringing in strategic interactions at all), then progressively less by adding more fidelity.

In general, player reduction requires that payoffs not be too sensitive to unilateral deviations—that payoffs vary smoothly with the number of players deviating. Although it is easy to concoct games for which any player reduction yields arbitrarily bad results, the evidence presented here for FPSB and local-effect games indicates that for some natural classes of games of interest we can use player reduction to find approximate equilibria. As we see in Chapter 6, some games cannot be approximated at all without the drastic computational savings afforded by player reduction.

3.6 Analyzing Empirical Games

In this section we assume that, using the methods in the previous sections, we have generated an empirical payoff matrix. As discussed in Section 2.1, GAMBIT is the state of the art finite game solver and a natural candidate for solving games given reasonably sized payoff matrices. In the next subsection we discuss GAMBIT's key shortcoming for this purpose: its inability to exploit symmetry. We then discuss additional game solving methods that do. None of these methods are original; however, to our knowledge we are the first to apply replicator dynamics (Section 3.6) as a solution method for large games [Wellman *et al.*, 2003c].

Existing Game Solvers and Symmetry

GAMBIT proceeds by iteratively eliminating strongly dominated strategies and then can apply a myriad of algorithms for solving the game, most notably the simplicial subdivision algorithm [McKelvey and McLennan, 1996] to enumerate the complete set of Nash equilibria. GAMBIT also has algorithms for solving extensive form games but we do not employ them in this thesis.

The problem with using GAMBIT in its current implementation is that it cannot take advantage of symmetry in a game. As noted in Section 1.2, a symmetric n-player game with S strategies has $\binom{n+S-1}{n}$ different profiles. Without exploiting symmetry, the payoff matrix requires S^n cells (each with n payoffs). This entails a huge computational cost just to store the payoff matrix. For example, for a 5-player game with 21 strategies, the payoff matrix needs over four million cells compared with 53 thousand when symmetry is exploited. Even for n and S in single digits, this difference quickly chokes GAMBIT on games that methods exploiting symmetry have no trouble with. For example, in many experiments we have run on 5-player, 5-strategy games, GAMBIT took hours or sometimes days to find all the Nash equilibria when it could find them at all. Asking GAMBIT to find only a sample equilibrium can be much faster. Still, there are games that GAMBIT fails to even load into memory that methods that do exploit symmetry can solve.

Two methods for searching for Nash equilibria that are readily extended to exploit symmetry are function minimization and replicator dynamics. We employ both of these methods in subsequent chapters to find sample equilibria.

Solving Games by Function Minimization

By Definition 3.2, $\varepsilon()$—a function from mixed strategy profiles to the reals—equals zero at s if and only if profile s is a Nash equilibrium. Since $\varepsilon()$ is bounded below by zero, we can apply function minimization techniques to solve games: the global minimum of $\varepsilon()$ corresponds to a Nash equilibrium. The problem with minimizing $\varepsilon()$ however is that it is neither continuous nor differentiable. The following function is continuously differentiable, and still retains the key properties of $\varepsilon()$—bounded below by zero and equal to zero only at Nash equilibria[15]—that allow us to find equilibria by globally minimizing it [McKelvey and McLennan, 1996].

$$f(s) = \sum_{s' \in S} \max[0, u(s', s) - u(s, s)]^2,$$

Following Walsh *et al.* [2002], we modify $f()$ (compared to $\varepsilon()$) in that its domain is the set of mixed strategies rather than mixed strategy profiles. In other words, we are limiting the search to symmetric equilibria (see Appendix C), which vastly speeds it up. We can search for the root of f using the Amoeba algorithm [Press *et al.*, 1992], a procedure for nonlinear function minimization based on the Nelder-Mead method [Nelder and Mead, 1965]. For our experiments in Chapter 5 we use an adaptation of Amoeba developed by Walsh *et al.* [2002] for finding symmetric mixed-strategy equilibria in symmetric games.

[15] This is because $f(s)$, like $\varepsilon(s)$, is positive iff any pure strategy is a strictly better response than s is to itself.

Solving Games by Replicator Dynamics

In his original exposition of the concept, Nash [1950] suggested an evolutionary interpretation of (what we now call) the Nash equilibrium. That idea can be codified as an algorithm for finding equilibria by employing the *replicator dynamics* formalism introduced by Taylor and Jonker [1978] and Schuster and Sigmund [1983].[16] Replicator dynamics refers to a process by which a population of strategies—where population proportions of the pure strategies correspond to mixed strategies—evolves over generations[17] by iteratively adjusting strategy populations according to performance with respect to the current mixture of strategies in the population. When this process reaches a fixed point, every pure strategy that has not died out is performing equally well given the current strategy mixture. Weibull [1995] shows that for two-player, two-strategy games, fixed points of a broad class of replicator processes are Nash equilibria if neither strategy is extinct. For n-player games, the set of fixed points that are locally asymptotically stable (all states sufficiently close converge to the same state) are a subset of the set of Nash equilibria [Friedman, 1991].

Although a more general form of replicator dynamics can be applied to nonsymmetric games [Gintis, 2000], it is particularly suited to searching for symmetric equilibria in symmetric games.

To implement this approach, we choose an initial population proportion for each pure strategy and then update them in successive generations so that strategies that perform well increase in the population at the expense of low-performing strategies. The proportion $p_g(s)$ of the population playing strategy s in generation g is given by

$$p_g(s) \propto p_{g-1}(s) \cdot (EP_s - W),$$

where EP_s is the expected payoff for pure strategy s against $n-1$ players all playing mixed strategies according to the population proportions, and W is a lower bound on payoffs (e.g., the minimum value in the payoff matrix). To calculate the expected payoff EP_s from the payoff matrix, we average the payoffs for s in the profiles consisting of s concatenated with every profile of $n-1$ other agent strategies, weighted by the probabilities of the other agents' profiles. The probability of a particular profile $\langle n_1, \ldots, n_S \rangle$ of n agents' strategies, where n_s is the number of players playing strategy s, is

$$\frac{n!}{n_1! \cdots n_S!} \cdot p(1)^{n_1} \cdots p(S)^{n_S}. \tag{3.10}$$

(This is the multinomial coefficient multiplied by the probability of a profile if order mattered.) Figure 3.7 shows an example of the replicator dynamics evolving to a particular symmetric mixed equilibrium in an SAA game (confer Chapter 5).

If the population update process reaches a fixed point, then the final population proportions are a candidate mixed strategy equilibrium. We verify directly that the candidate is in equilibrium by checking that the ε is zero or on par with the tolerance of the replicator dynamics calculations (typically 10^{-10}). In other words, we verify that the that the evolved strategy is a best response to itself. In all the experiments reported in this thesis, this process indeed reaches a fixed point and these fixed points always correspond to Nash equilibria. However, we have found examples for which the replicator dynamics do not converge and the population proportions cycle.

[16] Weibull [1995] attributes the the term “replicator” to Dawkins [1976].

[17] Instead of using discrete generations, the process can be defined continuously by a system of differential equations.

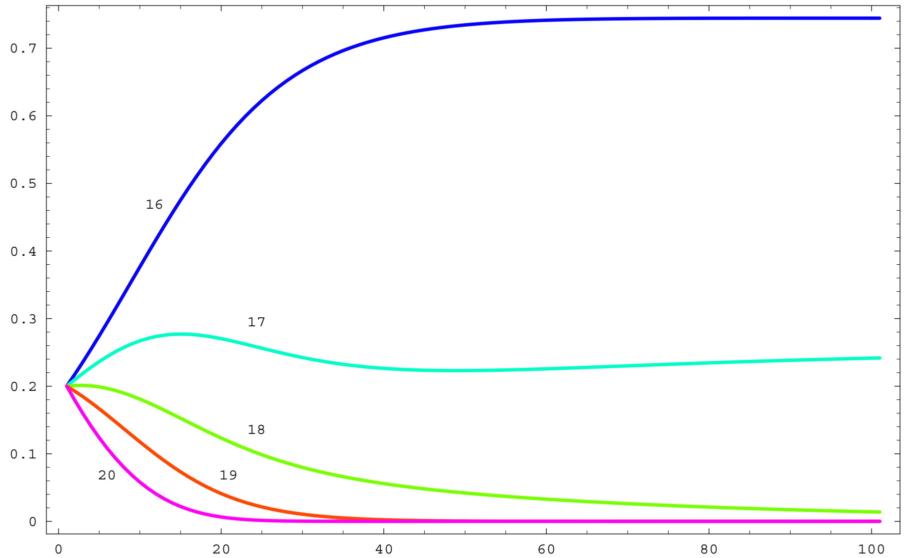

Figure 3.7: Replicator dynamics for an SAA game with 5 players and 5 strategies $\{16, \ldots, 20\}$ evolving in about 100 generations to a symmetric mixed equilibrium of all agents playing strategy 16 with probability 0.754 and 17 otherwise.

3.7 Interleaving Analysis and Estimation

Since estimating cells of a payoff matrix tends to be far more compute-intensive than finding Nash equilibria of a given payoff matrix, we discuss ways to use equilibrium analysis as a guide for selective sampling of profiles.

In related work, Walsh *et al.* [2003] show one way to apply this idea, interleaving the Monte Carlo simulation with the Nash equilibrium computation to concentrate sampling on profiles for which more accurate payoff estimates will yield better estimates of the Nash equilibria of the underlying game, allowing substantial reduction in the number of simulations needed to estimate a payoff matrix.

Their approach is inspired by information theory in that they choose the next profile to sample based on which has the greatest potential to change the maximum likelihood estimate for a Nash equilibrium. Note that this and other selective profile sampling techniques are complementary to the Monte Carlo variance reduction techniques of Section 3.4 which entails intelligent sampling of the underlying type distribution to speed up estimation of expected payoffs for a particular profile.

We have explored a related approach using a variation of replicator dynamics in which we simultaneously gather sample payoffs to form the estimated payoff matrix and evolve towards equilibrium. We start with an initial set of population proportions for each pure strategy. Then, as before, we repeatedly sample from the preference distribution and simulate the game to produce sample payoffs. However, now strategies are randomly drawn to participate according to their population proportions. Then, after a relatively small number of samples—long before we have confidence that they are good estimates of the expected payoffs—we apply the replicator dynamics using the realized average payoffs.[18] Then, given the new population proportions, we iterate, calculating a sequence of new generations, except that for each generation we retain the accumulated estimate of average payoffs from previous generations, and aggregate the old and new sample information. The iteration of generations continues until the population proportions are stationary with respect to the replicator dynamics.

The above procedure straightforwardly accumulates a statistically precise estimate of the expected payoff matrix. However, the sampling is biased: strategies that are more successful (and thus more highly represented in the population) are sampled more often. We conjecture that this approach can be more efficient than uniform sampling, especially for problems with a large number of permissible strategies. Since many of them will likely not be present in a Nash equilibrium—that is, their population fractions will converge to zero—extensive sampling to lower the standard error of the estimated payoff would be wasteful.

3.8 Sensitivity Analysis

Since our game analysis is based on payoff matrices estimated by repeated simulation of games, an important question is whether the equilibria we find are robust or if they would change with further sampling. Each expected payoff in the matrix is estimated (the maximum-likelihood estimate) by the sample mean but we need statistical confidence bounds on our estimate. More generally, we seek a probability distribution for the true expected payoff, representing our belief.

Estimating the Expected Value of a Random Variable

Since we have no a priori information about the distribution of the payoffs, our estimates of them are based only on our sampling. By the Central Limit Theorem,[19] the mean of a sample from any distribution converges to a normal distribution as the number of samples grows. For small sample sizes (less than 30) it is considered better to use a t-distribution. A corollary of Fisher's Theorem tells us that for samples from a normal distribution with true mean μ and sample size n with sample mean $\overline{X}$ and sample variance S^2, $(\overline{X} - \mu)/\sqrt{S^2/n}$ has a t-distribution with $n - 1$ degrees of freedom. Even when the underlying distribution is not normal, using such a t-distribution tends to be a better estimate.[20] Henceforth, I will refer to the distribution of the sample mean as normal even though we use the more faithful t-distribution for small sample sizes in our analysis. The key assumption, common in statistics, is that this sample mean distribution represents a reasonable belief for the underlying true mean—in our case, the true expected payoff. Figure 3.8 shows an example of how this assumption is reasonable even for highly non-normal distributions.

Approximating a Distribution over Payoff Matrices

Given a normal distribution for each of the expected payoffs in the payoff matrix we seek a distribution representing our belief of the space of possible payoff matrices. This problem is simplified

[18] Monte Carlo variance reduction techniques can also be applied here.

[19] Confer any statistics textbook, e.g., Rice [1995].

[20] To test this common presumption, we performed our own experiment. Using a class of highly non-normal distributions (0/1 Bernoulli trials with particular success probabilities) we repeated the following steps: (1) pick a distribution from the class (knowing the mean μ), (2) gather n samples from it, (3) use a proper scoring rule (quadratic) to score the estimated pdf of the sample mean, using a t-distribution and a normal. The steps were repeated for many underlying distributions from the class (with μ varying from 0 to 1) and for many different sample sizes. We conclude that a t-distribution indeed outperforms a normal distribution, until the sample size grows beyond 30 where the difference begins to vanish.

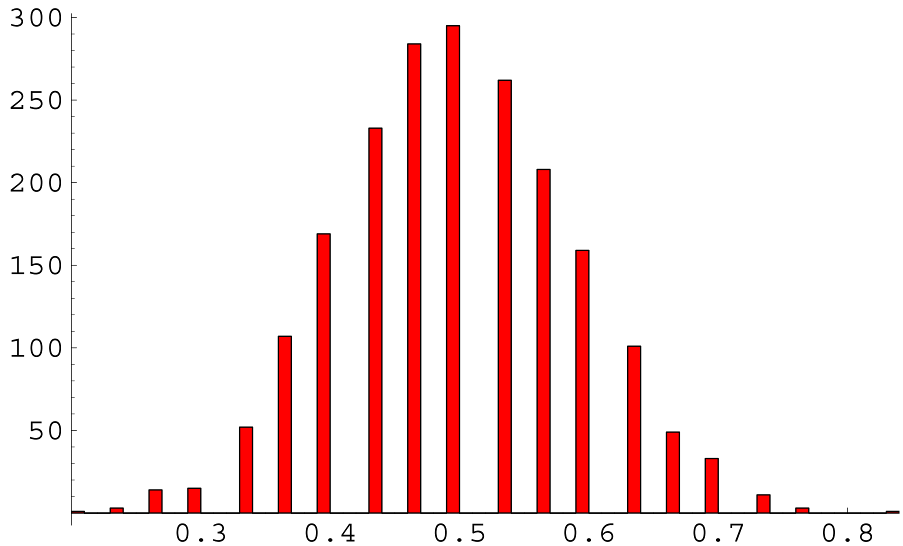

Figure 3.8: Histogram of the mean of 30 samples of a starkly bimodal distribution (0/1 Bernoulli trials with success probability 1/2—i.e., the density function has all its mass concentrated on 0 and 1).

by the fact that there is no covariance between payoffs in different cells. This is because they are functions of distinct vectors of independent random variables and thus must be independent. But payoffs within a cell—those corresponding to different strategies in the same profile—do covary in general. For example, in FPSB, the lower an agent's type, the less likely it is to win (the lower its expected payoff) with corresponding benefit to everyone else. We typically ignore this covariance and thus overestimate the variance of individual payoffs. This means that our sensitivity analysis is conservative in recommending additional samples (erring on the side of recommending more than necessary).

We compute the empirical payoff matrix distribution by collecting statistics from runs of the game simulator.[21] We can then aggregate samples within a profile where allowed by symmetry. For example, suppose we have simulated 1000 games for the profile $\langle 1, 1, 1, 2, 2\rangle$ in a 5-player game. By definition of a symmetric game, the first three expected payoffs (and the last two) must be the same. In fact, from the 1000 games we have 3000 instances of strategy 1 playing against two others playing 1 and two playing 2. Since samples from the same game are correlated, we cannot treat these as 3000 distinct instances[22] but we can nonetheless improve the estimates by replacing each sample payoff for the first three players by the average of the three (and similarly for the last two).

[21] It suffices to maintain a running total of the number of payoffs sampled, the sum, and the sum of the squares of the payoffs. From these summary statistics we can compute the mean and variance (and of course the sample size)—that is, the first and second moments.

[22] Though in our sensitivity analysis in Section 5.3 we did. Nonetheless, we believe the measure is still conservative and that even without conservative sample aggregation, the results are not suspect. This is consistent with our experience in that most often when our sensitivity analysis tells us we need more samples, they turn out not to have been necessary. In fact, sensitivity analysis has to indicate extreme sensitivity before it to turns out that with enough additional samples, the equilibria change. In other words, our informal observation is that this form of sensitivity analysis tends to report wider variances in equilibria than there are.

This shows that we are getting more than 1000 samples' worth of payoffs for the 1000 games and using 1000 as the sample size will again ensure that we only overestimate the variance of the payoff matrix distribution.

Assessing Sensitivity

Given a distribution over payoff matrices, we can infer a distribution over functions of the payoff matrix. One such function of interest is of course the set of Nash equilibria, or a sample equilibrium found by a particular solution method. Another is the ε for a particular profile (such as a candidate equilibrium).

Confidence Bounds on Mixture Probabilities

If enough samples from the payoff matrix distribution all yield the same equilibrium results then we would conclude that our results are insensitive to sampling noise. One way to operationalize that criterion is to repeatedly sample payoff matrices, compute one or all symmetric equilibria, and observe the distributions for the probability mixtures. Figure 3.9 shows an example of such a sensitivity measure from an SAA game (confer Chapter 5). In that example we can safely conclude that an equilibrium of the underlying restricted game involves playing strategy 16 with high probability, strategy 17 with low probability, and the other strategies with zero to very low probabilities. Throughout our experiments in subsequent chapters our experience has borne out the claim that this sensitivity measure is conservative. Equilibria judged robust by this measure do not tend to change with additional samples. In fact, only when this measure indicates extreme sensitivity do we find that additional simulation causes the equilibrium results to change.

Confidence Bounds on Epsilon

Unfortunately, we often do not have the simulation time needed to get the sample variances low enough to be able to say anything at all about the likely ranges of mixture probabilities in equilibrium.[23] So as an alternative sensitivity assessment, we turn to our ε metric and compute, for a candidate equilibrium profile s, an empirical distribution for $\varepsilon(s)$ in the same way that we found empirical distributions for the mixture probabilities above, by sampling many payoff matrices from our belief distribution and for each sample Γ_i computing $\varepsilon_{\Gamma_i}(s)$. If a fraction p of the probability mass is at zero then we can conclude with confidence p that we have found an actual equilibrium of the underlying restricted game. In the best case, p is near one and the conclusion is definitive. If not, we can at least give probabilistic bounds on ε and make concrete the claim that s approximates an equilibrium to the underlying (restricted) game. Figure 3.10 shows an empirical pdf (histogram) for ε of a candidate equilibrium based on an empirical estimate of a reduced TAC game (confer Chapter 6).

3.9 Related Work

Several researchers have addressed game-theoretic questions in a similar empirical vein to the approach presented in this chapter [Kephart *et al.*, 1998; Armantier *et al.*, 2000; Kephart and Greenwald, 2002; Walsh *et al.*, 2002]. Of particular note is the Trading Agent Competition (TAC) liter-

[23] We find that when we can identify a pure equilibrium it tends to be far more robust (less sensitive) than when only mixed equilibria are found.

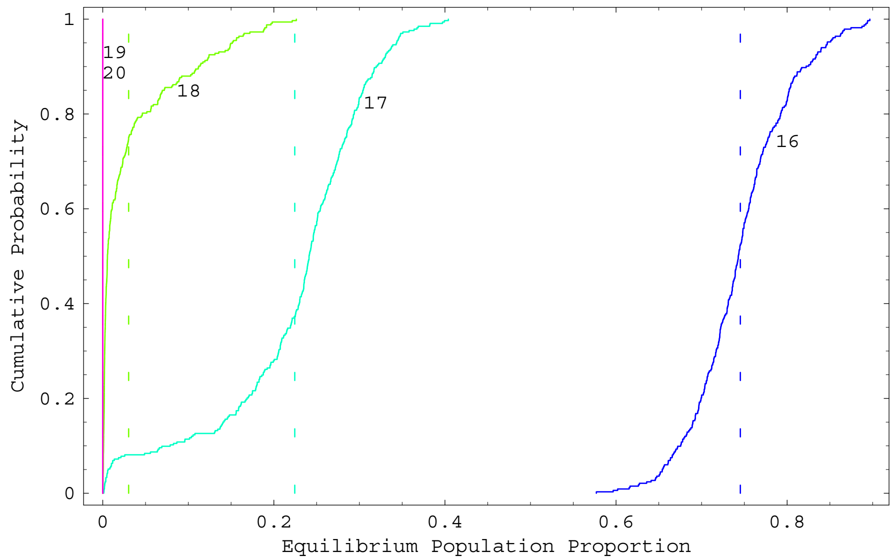

Figure 3.9: Sensitivity analysis for an SAA game with 5 players and 5 strategies $\{16, \ldots, 20\}$ showing the empirical cdfs for the equilibrium mixture probabilities. More vertical cdfs mean less sensitivity to sampling error. The payoff matrix was estimated by averaging the payoffs from 22 million samples for each of the 126 profiles.

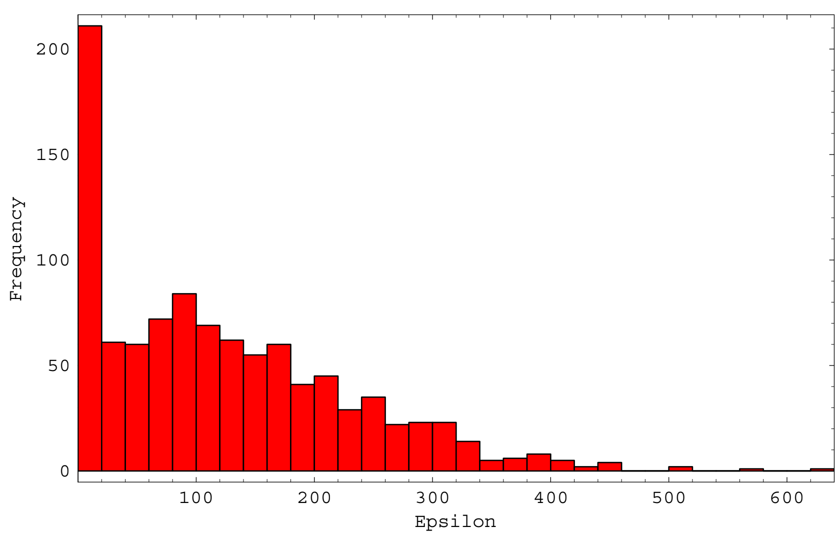

Figure 3.10: ε-sensitivity analysis for a 2-player reduced TAC game with 35 strategies showing the empirical pdf (histogram) for ε. The expected ε is 123 and the probability that the candidate is an actual equilibrium ($\varepsilon = 0$) is 15%. More probability mass near zero means a more robust profile. The partial payoff matrix was estimated from 14,000 samples spread (non-uniformly) over the 630 profiles, adjusted with control variates.

ature which, for the case of the travel-shopping domain, we discuss in Section 6.7. There has also been one study of note in the more recent supply-chain game introduced in TAC in 2003. In this domain, Wellman *et al.* [2005a] apply the basic approach of Section 3.3 along with control variates (Section 3.4) and sensitivity analysis (Section 3.8) to analyze strategic interactions in the game. As noted in Section 3.3, Armantier *et al.* [2000] define the concept of a Constrained Strategic Equilibrium (what we refer to as an equilibrium in a strategy-restricted game) and provide theoretical justification for an empirical game methodology. In more recent work, Armantier *et al.* [to appear] propose the core idea of estimating ε in the full game as a way to assess the quality of an equilibrium in the restricted game. Additionally, they apply a similar methodology to that described here to approximate equilibria in various auction games. As noted in Section 3.7, Walsh *et al.* [2003] have addressed the problem of reducing the computational cost of empirical payoff matrix generation. In Chapter 6 we apply a similar approach but in a rather ad hoc, manual way.

Despite the common use of symmetric constructions in game-theoretic analyses, the literature has not extensively investigated the general properties of symmetric games. One noteworthy exception is Nash's original paper on equilibria in non-cooperative games [Nash, 1951], which (in addition to presenting the seminal equilibrium existence result) considers a general concept of symmetric strategy profiles and its implication for symmetry of equilibria. We include a collection of results about symmetric games in Appendix C. Although GAMBIT does not yet exploit symmetry in games, Bhat and Leyton-Brown [2004] present a new game-solving algorithm based on a method by Govindan and Wilson [2003] which they extend to exploit symmetry and, especially, context-specific independencies between players' utility functions.

Evolutionary Search for Trading Strategies

Several prior studies have employed evolutionary techniques for the analysis or derivation of trading strategies. An early example was an evolutionary simulation among trading agents submitted to the Santa Fe Institute (SFI) Double Auction Tournament [Rust *et al.*, 1993].

The SFI Artificial Stock Market [Arthur *et al.*, 1997] has been used to investigate theories of trading behavior and market dynamics, incorporating genetic algorithms and other evolutionary mechanisms in versions of the model. This work is part of a growing literature in agent-based finance [LeBaron, 2000], much of which makes use of evolutionary techniques.

Price [1997] demonstrates the use of genetic algorithms (GAs) for a variety of standard industrial organization games (e.g., Bertrand and Cournot duopoly). In Price's approach, the GA serves as an optimization method, employed to derive a best response. For instance, his GA model for the duopoly games comprises populations of strategies for each producer, each updated separately according to GA rules. The search is co-evolutionary in the sense that fitness statistics are derived by joint sampling from the pair of populations.

Cliff [1998] applied GAs to evolve improved versions of his "ZIP" trading strategy for continuous double auctions. Improvement in his study is defined in terms of convergence to competitive equilibrium prices, as opposed to surplus for particular agents. The evolutionary search, therefore, is for a high-performing homogeneous trading policy, rather than a strategic equilibrium. In more recent work, Cliff [2003] expanded the search space to include a market-mechanism dimension, thus evolving a trading strategy in conjunction with an auction rule. As above, the GA's fitness measure is in terms of aggregate market performance, rather than individual profit.

Using a co-evolutionary approach similar to that of Price discussed above, Phelps *et al.* [2002] employ genetic programming to derive strategies for an electricity trading game studied by Nicolaisen *et al.* [2001]. They then extend the model to evolve auction rules in tandem with the trading strategies. Unlike Cliff, Phelps et al. evaluate fitness of the mechanism based on aggregate performance, while evolving trading strategies based on individual performance.

Byde [2002] evaluated a parameterized range of auction mechanisms, essentially equivalent to a one-sided version of k-double auctions [Satterthwaite and Williams, 1993]. For each scenario (auction setting and distribution of private and common values), he employs the GA to evolve a trading strategy, and evaluates the average revenue of the given auction with respect to a population of traders using that strategy.

Though sharing the same goal as this thesis (finding good trading strategies), the key difference between ours and the above evolutionary approaches is that we consider strategy generation game theoretically.

Chapter 4

Price Prediction for Strategy Generation in Complex Market Mechanisms

> IN WHICH *we propose various approaches to price prediction as the basis for strategy generation in interdependent auction environments.*

The previous chapter presents empirical methods for generating strategic guidance in games that are too complex for exact game-theoretic solutions. Here we apply the first step of that methodology—generating a space of candidate strategies—to two complex market games involving bidding under uncertainty in interdependent auctions: Simultaneous Ascending Auctions (SAA) and the Trading Agent Competition (TAC) travel-shopping game. We adopt the common approach of price prediction for finding strategies in such domains.

Our approach to price prediction in TAC is based on the concept of Walrasian price equilibrium, which we apply to SAA as well. After discussing this concept and its applicability to price prediction in general we consider the SAA and TAC domains separately. We present a general class of prediction-based strategies for SAA, parameterized by particular price predictions including Walrasian equilibrium prices and self-confirming predictions. (In Chapter 5 we demonstrate the robustness of the latter strategy.) Next, we present the details of our prediction-based hotel bidding strategy in TAC (other aspects of our TAC strategy are discussed in Chapter 6) and describe two methods for assessing the quality of predictions. Chapter 6 discusses our game-theoretic analysis of the TAC game.

4.1 Interdependent Auctions: TAC and SAA

The Trading Agent Competition travel-shopping game[1] pits eight travel agents against one another—each attempting to procure travel resources to put together trips for hypothetical clients. The preferences of these clients include preferred trip dates, premiums for better hotel rooms, and premiums for different types of entertainment tickets. These client preferences implicitly define a travel agent's value function over the available goods. The agent's task is to procure a set of goods that achieves maximum value at minimum cost. A key feature of the TAC game is the need for agents to participate in multiple simultaneous auctions.

[1] We describe the TAC game in more detail in Section 4.7. See also Wellman *et al.* [2003b].

This aspect of the Trading Agent Competition is characteristic of many real-world scenarios, such as bidders participating in concurrent auctions on eBay. To isolate the strategic issues involved in simultaneously bidding for related goods, we study a simpler and more generic domain that focuses on a set of identical and synchronized English ascending auctions. This is the Simultaneous Ascending Auctions (SAA) domain in which each agent participates concurrently in many auctions, one for each of the goods available. The auctions are independent except that no auction closes until bidding has stopped in all of them.

Complementarity and Substitutability

Complementarity is a key feature of both SAA and TAC. Complementarity manifests when an agent's value for a good is greater if it also obtains one or more other goods. For example, an airline passenger may wish to obtain two connecting segments to complete a trip. The airline, meanwhile, needs to obtain reservations for both a takeoff slot and a landing slot for each flight segment. If a pair (or in general a set) of goods are each worthless without the other, they are said to be perfect complements. Complementarity may also be partial: my demand for ski poles drops if the cost of skis becomes prohibitive, but I still may have value for one good without the other. In general, goods exhibit complementarity from the perspective of an agent when its valuation is *superadditive*. Given a set of goods X and valuation function,[2] $v\colon 2^X \to \mathbb{R}$, that assigns value to possible subsets of X, superadditive preference means that for any $Y \subseteq X$, $v(Y) \geq \sum_{i \in Y} v(\{i\})$. In other words, a bundle of goods is worth more than the sum of its parts.

When the inequality is reversed, the valuation is *subadditive* which implies substitutability between goods. For example, alternative and contemporaneous vacation packages are perfect substitutes—no additional value is derived from getting both. Two cameras may be partial substitutes if the bulk of an agent's value comes from getting one camera with some small additional value for having a second. An agent has *single-unit preference* iff for all $Y \subseteq X$, $v(Y) = \max_{i \in Y} v(\{i\})$. In other words, goods are perfect substitutes, though not necessarily 1:1 substitutes. For the case of subadditive utility and especially the extreme case of single-unit preference, finding optimal strategies is an easier problem.

In this thesis we focus on the more difficult case of superadditive preference—complementary goods. When an agent must bid for one resource with uncertainty about the market resolution of complements, its decision presents risky tradeoffs. By predicting eventual market prices, an agent can mitigate these risks.

4.2 Walrasian Price Equilibrium

A Walrasian price equilibrium refers to prices such that supply meets demand. It is also called a *competitive* price equilibrium since it constitutes prices that would be reached in a competitive economy, i.e., where all agents are price-takers.[3] Even though the economies we study are not competitive (except in the limit as the number of agents increases) we employ Walrasian competitive equilibria as estimates for the purposes of price prediction.

[2] In SAA, this value function (specified implicitly) defines an agent's type. The same is the case for TAC: the client preferences induce a valuation function over all possible subsets of travel resources. To fully specify a TAC agent's type also requires the initial endowment of goods. There is no such initial endowment in (our model of) SAA.

[3] Subject to additional assumptions, discussed below.

Consider a set of m goods[4] with prices denoted by the vector $\boldsymbol{p}$. There are n agents and each agent's demand for each good depends on the prices of all the goods, allowing in general for arbitrary complementarity and substitutability. We write $x_i^j(\boldsymbol{p})$ to denote an agent's demand for a good, specifically, the integer number of units of good i that agent j chooses to purchase given prices, $\boldsymbol{p}$, for all goods (including i). When only one unit of each good is available or desired, $x_i^j(\boldsymbol{p}) \in \{0,1\}\ \forall i,j,\boldsymbol{p}$. The vector of agent j's demands for all the goods is denoted $\boldsymbol{x}^j(\boldsymbol{p})$. *Aggregate demand* is simply the sum of agent demands,

$$\boldsymbol{x}(\boldsymbol{p}) = \sum_j \boldsymbol{x}^j(\boldsymbol{p}). \tag{4.1}$$

Note that an agent's valuation function defines its demand function. Specifically, an agent's demand vector $\boldsymbol{x}$ is that which (out of all possible demand vectors) maximizes valuation minus cost ($v(\boldsymbol{x}) - \boldsymbol{x}\cdot\boldsymbol{p}$).

Prices $\boldsymbol{p}$ constitute a Walrasian equilibrium iff aggregate demand equals aggregate supply for each good. In the case of exactly one of each of the m goods available, we have that in Walrasian equilibrium, $\boldsymbol{x}(\boldsymbol{p}) = \mathbf{1}$. To allow for multiple units of each good, the vector $\mathbf{1}$ is replaced with the aggregate supply vector, $\boldsymbol{y}$. Walras [1874] originally conceived of an iterative price adjustment mechanism—the *tatonnement* protocol—for searching for equilibrium prices [Arrow and Hahn, 1971]. Given a specification of aggregate demand and an initial guess $\boldsymbol{p}^0$, tatonnement iteratively computes a revised price vector according to:

$$\boldsymbol{p}^{t+1} = \boldsymbol{p}^t + \alpha^t(\boldsymbol{x}(\boldsymbol{p}^t) - \boldsymbol{y}), \tag{4.2}$$

where α^t is a learning parameter, strictly positive, that decreases in t.[5] The effect of the protocol is to nudge prices of under-demanded goods down and over-demanded goods up. If a fixed-point is reached then we have a price equilibrium.

General equilibrium theory develops technical conditions on agent preferences under which such an equilibrium can be guaranteed to exist, for the case of continuous goods [Mas-Colell *et al.*, 1995]. In our models, goods are discrete and for that case Bikhchandani and Mamer [1997] provide existence criteria. Single-unit preference is a sufficient condition for equilibrium existence, as well as other technical conditions, but for the general preference distributions we consider we are not guaranteed existence of price equilibria.[6] Indeed, there are examples in both domains of agent preferences that admit no price equilibria. For the convergence of tatonnement to a price equilibrium (when one exists) a sufficient condition is that preferences satisfy the property of *gross substitutes* [Kelso and Crawford, 1982]. Unfortunately, the complementarities in both TAC and SAA patently violate this property. Nonetheless, in both domains tatonnement typically converges quickly to at least an approximate price equilibrium (i.e., prices inducing relatively small imbalances of supply and demand).

---

[4] It is w.l.o.g. to consider m goods with exactly one unit available of each: if there are k units of one good they can be indexed and treated as distinct. Nonetheless, it is often convenient (e.g., in TAC) to allow for multiple units of each of the m goods (i.e., m good *types*) and so our treatment in this section allows for that case.

[5] For example, in TAC (see Section 4.9), after some experimentation we set $\alpha^t = 0.387 \cdot 0.95^t$.

[6] We do know that whenever a price equilibrium exists, the resulting allocations are guaranteed to be efficient [Bikhchandani and Mamer, 1997].

4.3 The Simultaneous Ascending Auctions (SAA) Domain

The formal specification of the Simultaneous Ascending Auctions (SAA) game [Cramton, 2005] includes a number of agents, n, a number of goods, m, and a type distribution that yields valuation functions, v_j, for the agents (plus Nature's type, needed for tie-breaking—see Section 3.3). Each separate auction, one for each good, may undergo multiple rounds of bidding. The bidding is synchronized so that in each round each agent submits a bid in every auction it chooses to bid in. At any given time, the *bid price* on good i is β_i, defined to be the highest non-repudiable bid b_i received thus far, or zero if there have been no bids. The bid price along with the current winner in every auction is announced at the beginning of each new round. To be admissible, a new bid must meet the *ask price*, i.e., the bid price plus a bid increment (which we take to be one w.l.o.g., allowing for scaling of the agent values): $b_i^{new} \geq \beta_i + 1$. If an auction receives multiple admissible bids in a given round, it admits the highest, breaking ties randomly. An auction is *quiescent* when a round passes with no new admissible bids, i.e., the new bid prices $\boldsymbol{\beta}^{new} = \boldsymbol{\beta}$ which become the final prices $\boldsymbol{p}$. When every auction is simultaneously quiescent they all close, allocating their respective goods per the last admitted bids. An agent's payoff—also referred to as its *surplus*—is then defined by the auction outcomes, namely, the set of goods it wins, X, and the final prices, $\boldsymbol{p}$:

$$\sigma(X, \boldsymbol{p}) \equiv v(X) - \sum_{i \in X} p_i. \tag{4.3}$$

Section 3.3 introduces the concept of a game simulator. As for any game, the input for the SAA simulator is the vector of agent types, $\boldsymbol{t}$, consisting for SAA of agent valuation functions plus Nature's type, and a strategy profile $\boldsymbol{s}$ (parameterizations for SAA strategies are described below). Agent strategies plus their types determine their bid actions and so we can simulate the protocol and compute the auction outcomes. We denote the auction outcomes by $X^j(\boldsymbol{t}, \boldsymbol{s})$ for the bundle of goods that agent j wins and $\boldsymbol{p}(\boldsymbol{t}, \boldsymbol{s})$ for the final prices, as a function of types and strategies. The game simulator for SAA is then:

$$SAA(\boldsymbol{t}, \boldsymbol{s})_j = \sigma(X^j(\boldsymbol{t}, \boldsymbol{s}), \boldsymbol{p}(\boldsymbol{t}, \boldsymbol{s})).$$

The Exposure Problem

Because no good is committed until all are, an agent's bidding strategy in one auction cannot be contingent on the outcome for another. Thus, an active agent desiring a bundle of complementary goods runs the risk that it will purchase some but not all goods in the bundle. This is known as the *exposure problem*, and arises whenever agents have complementarities among goods allocated through separate markets.

The exposure problem is a key motivation for designing mechanisms that take the complementarities directly into account, such as *combinatorial auctions* [Cramton *et al.*, 2005; de Vries and Vohra, 2003], in which the auction mechanism determines optimal packages based on agent bids over bundles. Although such mechanisms may provide an effective solution in many cases, there are often significant barriers to their application [MacKie-Mason and Wellman, 2005]. SAA-based auctions are even deliberately adopted, despite awareness of strategic complications [Milgrom, 2000]—most famously the US FCC spectrum auctions starting in the mid-1990s [McAfee and McMillan, 1996]. Simulation studies of scenarios based on the FCC auctions shed light on some strategic issues [Csirik *et al.*, 2001], as have accounts of some of the strategists involved [Cramton, 1995; Weber, 1997], but the game is still too complex to admit definitive strategic recommendations.

In general, markets for interdependent goods operating simultaneously and independently are ubiquitous, and yet auction theory to date [Krishna, 2002] has little to say about how one should bid in simultaneous markets with complementarities. In the next sections we propose the generation of candidate strategies based on price prediction and in Chapter 5 we apply the methodology from Chapter 3 to find and assess good instances of these strategies, as well as demonstrate their superiority over previously known strategies.

4.4 Strategies for SAA: Straightforward Bidding and Variations

If an agent knew the final prices of all m goods then its optimal strategy would be straightforward: bid on the subset of goods that maximizes its surplus at known prices. We define a class of bidding strategies based on *perceived* prices in which an agent selects the subset of goods that maximizes its surplus at those prices.

First, we define an agent's information state, $\boldsymbol{B}$, as the $t \times m$ history of bid prices revealed by the auctions as of the tth round. Any strategy is then a mapping from $\boldsymbol{B}$ to a bid vector—a bid in each of the m auctions where a zero bid is equivalent to not bidding. We denote the set of possible information states by $\boldsymbol{\mathcal{B}}$.

Definition 4.1 (Perceived-Price Bidder) *A* perceived-price bidder *is parameterized by a* perceived price function $\rho\colon \boldsymbol{\mathcal{B}} \to \mathbb{Z}_*^m$ *which maps the agent's information state, $\boldsymbol{B}$, to a (nonnegative, integer) perceived price vector, $\hat{\boldsymbol{p}}$. It computes the subset of goods*

$$X^* = \arg\max_X \sigma(X, \rho(\boldsymbol{B}))$$

breaking ties in favor of smaller subsets and lower-numbered goods.[7] *Then, given X^*, the agent bids $b_i = \beta_i + 1$ (the ask price) for the $i \in X^*$ that it is not already winning.*

A perceived-price bidding strategy is defined by how the agent constructs $\hat{\boldsymbol{p}}$ from its information state. In Section 4.5 we define functions, ρ, for perceived-price bidding strategies that map information state to *predicted* prices.

Straightforward Bidding

One example of a perceived-price bidder is the widely-studied straightforward bidding (SB) strategy.[8] An SB agent sets $\hat{\boldsymbol{p}}$ to *myopically perceived prices*: the bid price for goods it was winning in the previous round and the ask price for the others:

$$\rho_i(\boldsymbol{B}) = \hat{p}_i = \begin{cases} \beta_i & \text{if winning good } i \\ \beta_i + 1 & \text{otherwise,} \end{cases} \tag{4.4}$$

where $\boldsymbol{\beta}$ is the last (tth) row of $\boldsymbol{B}$, i.e., the current bid prices.

[7] *More precisely: when multiple subsets tie for the highest surplus, the agent chooses the smallest. If the smallest subset is not unique it picks the subset whose bit-vector representation is lexicographically greatest. (The bit-vector representation $\boldsymbol{\omega}$ of $X \subseteq \{1, \ldots, m\}$ has $\omega_i = 1$ if $i \in X$ and 0 otherwise. For example, the bit-vector representation of $\{1, 3\} \subseteq \{1, 2, 3\}$ is $\langle 1, 0, 1 \rangle$.)*

[8] We adopt the terminology introduced by Milgrom [2000]. The same concept is also referred to as "myopic best response", "myopically optimal", and "myoptimal" [Kephart *et al.*, 1998].

Straightforward bidding is a reasonable strategy in some environments, such as the special case of an SAA with no complementarities. When all agents have single-unit preference, and value every good equally (i.e., the goods are all 1:1 perfect substitutes), the situation is equivalent to a problem in which all buyers have an inelastic demand for a single unit of a homogeneous commodity. For this problem, Peters and Severinov [2001] show that straightforward bidding is a perfect Bayes-Nash equilibrium.

If agents have additive utility, i.e., $v(Y) = \sum_{i \in Y} v(\{i\})$, then they can treat the auctions as independent and in this case too, SB is in equilibrium. To see this, consider the case that all other agents are playing SB with additive preference. Then your bid in one auction does not affect your surplus in another. This implies the auctions can be treated independently and SB is a best response. We conjecture that SB is a Nash equilibrium in the more general subadditive case as well, but a definitive answer to this question awaits future work.

The degenerate SAA with $m = 1$, i.e., a single ascending auction, is strategically equivalent to a second-price sealed-bid auction [Vickrey, 1961].[9] For $m > 1$, however, even though additivity makes the auctions independent, the joint strategy space allows *threats* such as "if you raise the price on my good I will raise it on yours." These will then support demand-reduction equilibria, even in the additive case. Thus, although SB is a good strategy and is in equilibrium for at least some classes of environments without complementarities, it is not a dominant strategy in any reasonable class of SAA games.

Up to a discretization error, the allocation in an SAA with single-unit preference is efficient when agents follow straightforward bidding. It can also be shown [Bertsekas, 1992; Wellman *et al.*, 2001a] that the final prices will differ from the minimum unique equilibrium prices by at most $\min(m, n)$ times the bid increment. The value of the allocation, defined to be the sum of the bidder surpluses, will differ from the optimal by at most the bid increment times $\min(m, n)(1 + \min(m, n))$.

Unfortunately, none of these properties generalize to superadditive preference. The final SAA prices can differ from the minimum equilibrium price vector, and the allocation value can differ from the optimal, by arbitrarily large amounts [Wellman *et al.*, 2001a]. And most importantly, SB need not be a Nash equilibrium.

	$v(\{1\})$	$v(\{2\})$	$v(\{1,2\})$
Agent 1	5	5	5
Agent 2	0	0	8

Table 4.1: A simple problem illustrating the pitfalls of SB (Example 4.2).

Example 4.2 *There are two agents, with values for two goods as shown in Table 4.1. One admissible straightforward bidding path*[10] *leads to a state in which agent 2 is winning both goods at prices (4,3). Then, in the next round, agent 1 would bid 4 for good 2. The auction would end at this point, with agent 1 receiving good 2 and agent 2 receiving good 1, both at a price of 4.*

[9] Technically this equivalence applies to a strategically restricted version of the ascending auction which does not allow arbitrary bids above the ask price (and raises the ask price continuously rather than discretely). Otherwise there exist strategies (albeit pathological) to which SB is not a best response. For example, suppose my policy is to not bid more than $100 unless the bidding starts lower in which case I will keep bidding indefinitely. The best response to such a strategy requires *jump bidding*.

[10] *The progression of the SAA protocol depends on tie-breaking.*

In this example, SB leads to a result with total allocation value 5, whereas the optimal allocation would produce a value of 8. Adding goods and agents would enable constructing slightly more complex examples, magnifying the suboptimality to an arbitrary degree.

So we see that straightforward bidding fails to guarantee high quality allocations except in highly restricted problems. It is also easy to show that straightforward bidding is not an equilibrium strategy in general. Consider again Example 4.2. With SB agents, the mechanism reaches quiescence at prices (4,4). However, it is not rational for agent 2 to stop at this point. If, for example, it bid 5 for good 2, the auction would end (assuming agent 1 plays SB), and agent 2 would be better off, with a surplus of -1 rather than -4.

It is clear that SB is not a reasonable candidate for a general strategy in SAA. Recall that in Chapter 3 we generalized an awful strategy for FPSB—truthful bidding—by adding a shade parameter which, when set properly, yielded the unique symmetric Nash equilibrium of the game. We now show how a similarly simple parametric generalization to SB can address a key strategic shortfall.

Sunk-Awareness

We showed in Example 4.2 that in some problems agents following a straightforward bidding strategy may stop bidding prematurely. We now consider why SB is failing in this situation. In a given round, agents bid on the set of goods that maximizes their surplus at myopically perceived prices (current bid or ask prices). If none of the nonempty subsets of goods appear to yield positive net surplus, the agent chooses the empty set, i.e., it does not bid at all, because the alternative is to earn negative surplus. However, this behavior ignores outstanding commitments: the agent may already be winning one or more goods. If the agent drops out of the bidding, and others do not bid away the goods the agent already is winning, then its alternative surplus could be much worse than if it continued to bid despite preferring the empty bundle at current prices. In the case of an agent dropping out of the bidding on some goods in a bundle of perfect complements, its surplus is negative the sum of the bid prices for the goods in the bundle it gets stuck with. This failure of straightforward bidding is due to ignoring the true opportunity cost of not bidding.

We refer to this property of straightforward bidding as "sunk unawareness" [Reeves *et al.*, 2005]. SB agents bid as if the incremental cost for goods they are currently winning is the full price. However, since they are already committed to purchasing these goods (if another agent does not raise the bid price), the cost is sunk, and the incremental cost is zero.

Given this clear failure of straightforward bidding, we parameterize a family of perceived-price bidders (Definition 4.1) that permits agents to account to a greater or lesser extent for the true incremental cost of goods they are currently winning. We call this strategy "sunk aware". A sunk-aware agent bids as if the incremental cost for goods it is currently winning is somewhere on the interval of zero and the current bid price.

A sunk-aware agent generalizes SB's method for choosing its perceived price vector (Equation 4.4) with the parameter $k \in [0, 1]$:

$$\rho_i(\boldsymbol{B}) = \hat{p}_i = \begin{cases} k\beta_i & \text{if winning good } i \\ \beta_i + 1 & \text{otherwise.} \end{cases}$$

If $k = 1$ the strategy is identical to straightforward bidding. At $k = 0$ the agent is fully sunk aware, bidding as if it would retain the goods it is currently winning with certainty. Intermediate values are akin to bidding as if the agent puts an intermediate probability on the likelihood of retaining the goods it is currently winning. We treat as a special case sunk-aware agents with single-unit preference: regardless of their k they bid straightforwardly ($k = 1$) since for such agents SB is a no-regret strategy. In other words, all sunk-aware agents with single-unit preference are really SB agents.

The sunk-awareness parameter provides a heuristic for a complex tradeoff: the agent's bidding behavior changes after it finds itself exposed to the underlying problem (owning goods it may not be able to use). In Chapter 5, we report on our experiments to discover good settings of the sunk-awareness parameter.

Any perceived-price bidder (in particular SB and sunk-aware) is a degenerate form of a price predicting strategy—the agent bids as if it believes its perceived prices will be the final prices. We next consider strategies that explicitly use a model of the game, including the type distribution, to generate price predictions. This allows an agent to anticipate possible exposure, and adapt its bidding behavior for the anticipated risk.

4.5 Price Prediction Strategies for SAA

Whenever an agent has superadditive preference and chooses to bid on a bundle of size > 1, it may face exposure. Exposure in SAA is a direct tradeoff: bidding on a needed good increases the prospects for completing a bundle, but also increases the expected loss in case the full set of required goods cannot be acquired. A decision-theoretic approach would account for these expected costs and benefits, choosing to bid when the benefits prevail, and cutting losses in the alternative.

Agent	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	$\{1,2,3\}$
1	0	0	0	0	0	0	15
2	8	6	5	8	8	6	8
3	10	8	6	10	10	8	10

Table 4.2: Agent valuation functions for a problem illustrating the value of price prediction over SB and sunk-aware agents.

Example 4.3 *There are three agents with values for three goods as shown in Table 4.2. Agents 2 and 3 have single-unit preference. Agent 1 needs all three goods to obtain any value (the goods are perfect complements). If all three agents play SB, a possible outcome is that agent 3 wins the first good at 8, agent 1 wins the second at 5, and agent 2 wins the third at 4.*

The previous example (4.2) showed that SB may stop bidding prematurely. The sunk-aware strategy solves that problem, but now consider agent 1's plight in Example 4.3. It is caught by the exposure problem, stuck with a useless good and a surplus of -5. (Other tie-breaking choices result in different outcomes but all of them leave agent 1 exposed and with negative surplus.) Playing a fully sunk-aware strategy could bring its surplus as high as -1 but it would still of course fare better by not bidding at all.

The effectiveness of a particular strategy will in general be highly dependent on the characteristics of other agents in the environment. This observation motivates the approach of price prediction. We would prefer strategies that employ type distribution beliefs to guide bidding behavior, rather than relying only on current price information as in the sunk-aware strategies (including SB). Forming price predictions for the goods in SAA is a natural use for type distribution beliefs. In Example 4.3, suppose agent 1 could predict with certainty before the auctions start that the prices would total at least 16. Then it could conclude that bidding is futile, not participate, and avoid the exposure problem altogether. Of course, agents will not in general make perfect predictions. However, we find that even modestly informed predictions can significantly improve performance.

We now offer straightforward ways to improve on SB and sunk-awareness by using explicit price predictions for perceived prices. In the following subsections we present three easy-to-compute strategies that are instances of this approach.

Let $F \equiv F(\boldsymbol{B})$ denote a joint cumulative distribution function over final prices, representing the agent's belief given its current information state $\boldsymbol{B}$. We assume that prices are bounded above by a known constant, V. Thus, F associates probabilities with price vectors in $\{1, \dots, V\}^m$.

We next consider two ways to use prediction information to generate perceived prices. We first define a *point prediction* for perceived prices, $\boldsymbol{\pi}$, that anticipates possible exposure risks. Then we define a *distribution prediction* for perceived prices, $\boldsymbol{\Delta}$, that in addition adjusts for the degree to which the agent's current winning bids are likely to be sunk costs. (As with sunk-awareness, all price predicting agents with single-unit preference ignore their predictions and play SB.)

Point Price Prediction

Suppose the agent has (at least) point beliefs about the final prices that will be realized for each good. Let $\boldsymbol{\pi}(\boldsymbol{B})$ be a vector of predicted final prices. Before the auctions begin the price prediction is $\boldsymbol{\pi}(\varnothing)$, where $\varnothing$ is the empty set of bid information available pre-auctions.

The auctions in SAA reveal the bid prices each round. Since the auctions are ascending, once the current bid price for good i reaches β_i, there is zero probability that the final price p_i will be less than β_i. We define a simple updating rule using this fact: the current price prediction for good i is the maximum of the initial prediction and the myopically perceived price:

$$\pi_i(\boldsymbol{B}) \equiv \begin{cases} \max(\pi_i(\varnothing), \beta_i) & \text{if winning good } i \\ \max(\pi_i(\varnothing), \beta_i + 1) & \text{otherwise.} \end{cases} \tag{4.5}$$

Armed with these predictions, the agent plays the perceived-price bidding strategy (Definition 4.1) with $\rho(\boldsymbol{B}) \equiv \boldsymbol{\pi}(\boldsymbol{B})$. We denote a specific point price prediction strategy in this family by $\mathrm{PP}(\boldsymbol{\pi}^x)$, where x labels particular initial prediction vectors, $\boldsymbol{\pi}(\varnothing)$. Note that straightforward bidding is the special case of price prediction with the predictions all equal to zero: SB $= \mathrm{PP}(\mathbf{0})$. If the agent underestimates the final prices, it will behave identically to SB after the prices exceed the prediction. If the agent overestimates the final prices, it will stop bidding prematurely.

Distribution-Based Price Prediction

Strategies using additional information from the distribution F can at least weakly dominate strategies using only a prediction of the final price distribution mean (i.e., the expectation of F). We assume the agent generates $F(\varnothing)$, an initial, pre-auction belief about the distribution of final prices.

As with the point predictor, we restrict the updating in our distribution predictor to conditioning the distribution on the fact that prices must be bounded below by $\boldsymbol{\beta}$. Let $\Pr(\boldsymbol{p} \mid \boldsymbol{B})$ be the probability, according to F, that the final price vector will be $\boldsymbol{p}$, conditioned on the information revealed by the auction, $\boldsymbol{B}$. Then, with $\Pr(\boldsymbol{p} \mid \varnothing)$ as the pre-auction initial prediction, we define:

$$\Pr(\boldsymbol{p} \mid \boldsymbol{B}) \equiv \begin{cases} \dfrac{\Pr(\boldsymbol{p} \mid \varnothing)}{\sum_{\boldsymbol{q} \geq \boldsymbol{\beta}} \Pr(\boldsymbol{q} \mid \varnothing)} & \text{if } \boldsymbol{p} \geq \boldsymbol{\beta} \\ 0 & \text{otherwise.} \end{cases} \tag{4.6}$$

(By $\boldsymbol{x} \geq \boldsymbol{y}$ we mean $x_i \geq y_i$ for all i.) For (4.6) to be well defined for all possible $\boldsymbol{\beta}$ we define the price upper bounds such that $\Pr(V, \ldots, V \mid \varnothing) > 0$.

We now use the distribution information to implement a further enhancement to take sunk costs into account in a more decision-theoretic way than the sunk-aware agent. If an agent is currently not winning a good and bids on it, then the expected incremental cost of winning the good is the expected final price, with the expectation calculated with respect to the distribution F. If the agent is currently winning a good, however, then the expected incremental cost of winning that good depends on the likelihood that the current bid price will be increased by another agent, so that the first agent has to bid again to obtain the good. If, to the contrary, it keeps the good at the current bid, the full price is sunk (already committed) and thus should not affect incremental bidding. Based on this logic we define $\Delta_i(\boldsymbol{B})$, the expected *incremental* price for good i.

First, for simplicity we use only the information contained in the vector of marginal distributions, $(F_1, \ldots, F_m)$, as if the final prices were independent across goods. Define the expected final price conditional on the most recent vector of bid prices, $\boldsymbol{\beta}$:

$$E_F(p_i \mid \boldsymbol{\beta}) = \sum_{q_i=0}^{V} \Pr(q_i \mid \beta_i) q_i = \sum_{q_i=\beta_i}^{V} \Pr(q_i \mid \beta_i) q_i.$$

The expected incremental price depends on whether the agent is currently winning good i. If not, then the lowest final price at which it could is $\beta_i + 1$, and the expected incremental price is simply the expected price conditional on $p_i \geq \beta_i + 1$,

$$\Delta_i^{\mathrm{L}}(\boldsymbol{B}) \equiv E_F(p_i \mid \beta_i + 1) = \sum_{q_i=\beta_i+1}^{V} \Pr(q_i \mid \beta_i + 1) q_i.$$

If the agent is winning good i, then the incremental price is zero if no one outbids the agent. With probability $1 - \Pr(\beta_i \mid \beta_i)$ the final price is higher than the current price, and the agent is outbid with a new bid price $\beta_i + 1$. Then, to obtain the good to complete a bundle, the agent will need to bid at least $\beta_i + 2$, and the expected incremental price is

$$\Delta_i^{\mathrm{W}}(\boldsymbol{B}) = (1 - \Pr(\beta_i \mid \beta_i)) \sum_{q_i=\beta_i+2}^{V} \Pr(q_i \mid \beta_i + 2) q_i.$$

The vector of expected incremental prices is then defined by

$$\Delta_i(\boldsymbol{B}) = \begin{cases} \Delta_i^{\mathrm{W}}(\boldsymbol{B}) & \text{if winning good } i \\ \Delta_i^{\mathrm{L}}(\boldsymbol{B}) & \text{otherwise.} \end{cases}$$

The agent then plays the perceived-price bidding strategy (Definition 4.1) with $\rho(\boldsymbol{B}) \equiv \boldsymbol{\Delta}(\boldsymbol{B})$. We denote the strategy of bidding based on a particular distribution prediction by $\mathrm{PP}(F^x)$, where x labels various distribution predictions, $F(\varnothing)$.

4.6 Methods for Predicting Prices in SAA

Definition 4.1 parameterizes the class of perceived-price bidding strategies and in Section 4.5 we define point price and distribution-based prediction methods that construct perceived prices from the agent's information state. The point and distribution predictors are classes of strategies parameterized by the choice of initial prediction—a vector of predicted final prices in the case of the point predictor, or, more generally, a distribution of final prices for the distribution predictor. We now present several ways to obtain an initial prediction.

Walrasian Equilibrium for Point and Distribution Prediction

We describe in Section 4.2 methods for finding approximate Walrasian price equilibria given a set of m goods and agent valuation functions over those goods. But an agent in SAA knows only its own valuation function. (Recall that knowing an agent's type—i.e., valuation function—is tantamount to knowing its demand function.) This presents a problem for an agent that wishes to use Walrasian equilibrium for price prediction but knows only the distribution from which other agents' types are drawn.

Given a distribution of agent types we can generalize the price equilibrium calculation in two ways to allow for probabilistic knowledge of the aggregate demand function. The first is to find the *expected price equilibrium* (EPE): the expectation (over the type distribution) of the Walrasian price equilibrium vector. The most straightforward way to estimate this is Monte Carlo simulation, sampling from the type distribution. For any particular sampled type, the demand function $\boldsymbol{x}$ is determined and the tatonnement protocol can be applied. Repeated sampling of types and application of tatonnement with each sample yields a crude Monte Carlo estimate of the expected price equilibrium.

An alternative (which we may prefer for computational reasons) to estimating a price equilibrium in the face of probabilistic demand is the *expected demand price equilibrium* (EDPE): the Walrasian price equilibrium with respect to expected aggregate demand. In other words, we calculate or estimate the expected demand function and apply the tatonnement protocol once. We calculate expected demand analytically when possible; otherwise we can estimate it by Monte Carlo simulation, again sampling from the type distribution.

Either of these generalized Walrasian price equilibrium methods can be applied to generate point predictions. We denote the expected price equilibrium point predictor by $\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{EPE}})$ and the expected demand price equilibrium point predictor by $\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{EDPE}})$. The method of expected price equilibrium can also be straightforwardly generalized—by tracking the empirical distribution of price equilibria instead of just average prices—to the case of distribution predictors. This predictor is denoted $\mathrm{PP}(F^{\mathrm{EPE}})$. For all the Walrasian predictors applied to SAA the prediction is based purely on expected demand for all n agents. An agent employing this strategy could likely improve the prediction by instead adding its own known demand with the expected demand of the other $n-1$ agents. We do so for the TAC domain (Section 4.9). In Chapter 5 (Sections 5.6 and 5.7) we report on SAA experiments that include Walrasian price predictors.

Predictions from Historical Data

One simple method for generating an initial prediction is to run simulations (equivalently, observe a history of games) to produce an empirical price distribution for a given strategy profile. In other words, we run the game simulator repeatedly, tracking not payoffs but final prices, similar to the tatonnement-sampling approach to estimating equilibrium price distributions. For a point price prediction we compute average final prices, and for a distribution-based prediction we compute final price histograms. A price prediction strategy can then be specified by the type of predictor (point vs. distribution) and a strategy profile from which to glean a distribution of final prices.

Baseline Prediction

As a noteworthy special case of the above, our baseline prediction is the distribution of final prices resulting from all SB agents. We denote the baseline point predictor $\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{SB}})$ and the baseline distribution predictor $\mathrm{PP}(F^{\mathrm{SB}})$. In fact, any strategy profile can be the basis for such a simulation-based predictor, including other predictors. We next consider a prediction strategy based on simulations of itself.

Self-Confirming Predictions

We now define the concept of self-confirming predictions for final prices in SAA, first for the case of point prediction and then the more general case of distribution prediction.

Definition 4.4 (Self-Confirming Point Price Prediction) *Let Γ be an instance of an SAA game. The prediction $\boldsymbol{\pi}$ is a* self-confirming prediction for Γ *iff $\boldsymbol{\pi}$ is equal to the expectation (over the type distribution) of the final prices when all agents play* $\mathrm{PP}(\boldsymbol{\pi})$.

In other words, if all agents use predictions, then the self-confirming predictions are those that on average *are correct*. We denote the self-confirming prediction vector by $\boldsymbol{\pi}^{\mathrm{SC}}$ and the self-confirming point prediction strategy by $\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{SC}})$.

To find approximate self-confirming predictions, we follow a simple iterative procedure. First, we initialize the predicting agents with some prediction vector (e.g., all zero) and simulate many games with the all-predict profile. When average prices obtained by these agents are determined, we replace the prediction vector with the average prices and repeat. When this process reaches a fixed point, we have the self-confirming prediction, $\boldsymbol{\pi}^{\mathrm{SC}}$. In Figure 4.1 we show the convergence to the five self-confirming prediction prices for a particular SAA environment with five agents and five goods. Within 30 iterations the prices have essentially converged, although there is some persistent oscillation. We found that by reseeding the prediction vector (with the averages around which the prices are oscillating) the process converged immediately to a more precise fixed point, which we use as $\boldsymbol{\pi}^{\mathrm{SC}}$.

We now define the concept of a self-confirming *distribution* of final prices in SAA.

Definition 4.5 (Self-Confirming Price Distribution) *Let Γ be an instance of an SAA game. The prediction F is a* self-confirming price distribution for Γ *iff F is the distribution of prices resulting when all agents play bidding strategy* $\mathrm{PP}(F)$.

A prediction is *approximately self-confirming* if the definition above is satisfied for some approximate sense of equivalence between the outcome and prediction distributions. We have developed in detail [Osepayshvili *et al.*, 2005] a procedure for deriving self-confirming distribution predictions. In the same paper we show that self-confirming predictions are not guaranteed to exist. We do this via a carefully constructed type distribution that serves as a counterexample. We conjecture, however, that for reasonably diffuse type distributions, self-confirming predictions will exist (as they do for the environments we study in the next chapter).

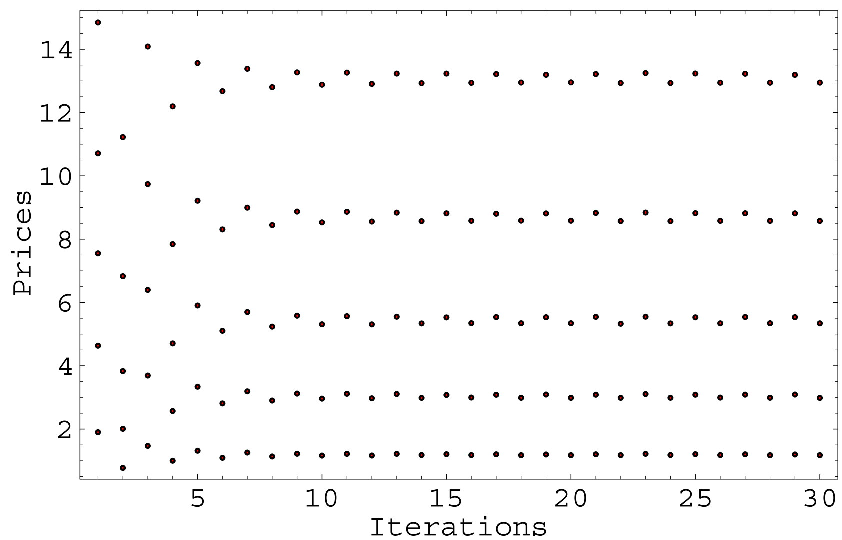

Figure 4.1: Convergence to a self-confirming price-prediction vector, starting with initial prediction that all prices would be zero. The prices at each iteration are determined by 500 thousand simulated games.

The key feature of self-confirming predictions is that agents make decisions based on predictions that turn out to be correct with respect to the type distribution and the strategies played. Since agents are optimizing for these predictions, we might reasonably expect the strategy to perform well in an environment where its predictions are confirmed.

The actual joint distribution will in general have dependencies across prices for different goods. We are also interested in the situation in which if the agents play a strategy based just on marginal distributions, that resulting distribution has the same marginals, despite dependencies.

Definition 4.6 (Self-Confirming Marginal Distribution) *Let* Γ *be an instance of an SAA game. The prediction* $F = (F_1, \ldots, F_m)$ *is a vector of* self-confirming marginal price distributions for Γ *iff for all* i*,* F_i *is the marginal distribution of prices for good* i *resulting when all agents play bidding strategy* $\mathrm{PP}(F)$ *in* Γ*.*

Note that the confirmation of marginal price distributions is based on agents using these predictions as if the prices of goods were independent. However, we consider these predictions confirmed in the marginal sense as long as the results agree for each good separately, even if the joint outcomes do not validate the independence assumption. It is self-confirming marginal distribution predictors we analyze in Chapter 5. We denote this strategy by $\mathrm{PP}(F^{\mathrm{SC}})$.

4.7 The Trading Agent Competition (TAC) Domain

Having defined price prediction strategies for SAA, we turn now to the Trading Agent Competition domain.[11] TAC presents a travel-shopping task where traders assemble flights, hotels, and entertain-

[11] TAC comprises two competitions: TAC Travel (also called TAC Classic)—the travel-shopping game—and TAC SCM, a supply-chain management scenario introduced in 2003. I focus exclusively on TAC Travel in this thesis.

ment into trips for a set of eight probabilistically generated clients. Clients are described by their preferred arrival and departure days (pa and pd), the premium (hp) they are willing to pay to stay at the "Towers" (T) hotel rather than "Shanties" (S), and their valuations for three different types of entertainment events. The agents' objective is to maximize the value of trips for their clients, net of expenditures in the markets for travel goods.[12]

Each of the three types of travel goods are bought and sold in distinct types of markets:

Flights. A feasible trip includes round-trip air, which consists of an inflight day i and outflight day j, $1 \leq i < j \leq 5$. Flights in and out each day are sold independently, at prices determined by a stochastic process. The initial price for each flight is $\sim U[250, 400]$, and follows a random walk thereafter with a bias determined by a hidden parameter chosen randomly at the start of the game. Specifically, at the start of each game, a hidden parameter x is chosen $\sim U[-10, 30]$. Define $x(t) = 10 + (t/9{:}00)(x - 10)$. Every 10 seconds thereafter, given elapsed time t, flight prices are perturbed by a value chosen uniformly, with bounds $[lb, ub]$ determined by

$$[lb, ub] = \begin{cases} [x(t), 10] & \text{if } x(t) < 0 \\ [-10, 10] & \text{if } x(t) = 0 \\ [-10, x(t)] & \text{if } x(t) > 0. \end{cases} \tag{4.7}$$

Hotels. Feasible trips must also include a room in one of the two hotels for each night of the client's stay. There are 16 rooms available in each hotel each night, and these are sold through ascending 16th-price auctions. When the auction closes, the units are allocated per the 16 highest offers, with all bidders paying the price of the lowest winning offer. Each minute, the hotel auctions issue *quotes*, indicating the 16th- (ASK) and 17th-highest (BID) prices among the currently active unit offers. Each minute, one of the hotel auctions is selected at random to close, with the others remaining active and open for bids. Note that, especially at the beginning of the game before any hotel auctions have closed, the hotel market bears distinct similarity to SAA.

Entertainment. Agents receive an initial random allocation of entertainment tickets (indexed by type and day), which they may allocate to their own clients or sell to other agents through continuous double auctions. We treat entertainment trading as a black box in this thesis.

A feasible client trip r is defined by an inflight day in_r, outflight day out_r, and hotel type (H_r, which is 1 if T and 0 if S). Trips also specify entertainment allocations, but for purposes of this thesis we summarize expected entertainment surplus ϕ as a function of trip days. (The full description of our agent [Cheng *et al.*, 2005] includes details on $\phi(r)$.) The value of this trip for client c (with preferences pa, pd, hp) is then given by

$$v_c(r) = 1000 - 100(|pa - in_r| + |pd - out_r|) + hp \cdot H_r + \phi(r).$$

At the end of a game instance, the TAC server calculates the optimal allocation of trips to clients for each agent, given final holdings of flights, hotels, and entertainment. The agent's game score is its total client trip utility, minus net expenditures in the TAC auctions.

[12] Full details of the game rules are documented at `http://www.sics.se/tac`.

4.8 Price Prediction in TAC

TAC participants recognized early on the importance of accurate price prediction in overall performance [Stone and Greenwald, 2005]. As in SAA, agents in TAC have an incentive to avoid the exposure problem when bidding for complementary hotels. Compounding this is the flights—also complementary goods—which increase in cost throughout the game. Thus, agents have an incentive to commit to trips early (by buying flights). The better their predictions of final hotel prices the more profitable their trip selections will be (see Section 4.11).

Given the importance of price prediction, it is not surprising that TAC researchers have explored a variety of approaches. Stone *et al.* [2002] noted a diversity of price estimation methods among TAC agents in the second year of the competition. After the third year of competition in 2002, we performed an extensive survey and comparison of prediction techniques [Wellman *et al.*, 2004]. (I include the conclusion of that study in Section 4.11.) Our own TAC agent, Walverine,[13] introduced a prediction method quite distinct from those reported previously, namely, predicting Walrasian equilibrium prices.

In the remainder of this chapter, we consider the price prediction subtask of TAC.[14] Although interesting subtasks tend not to be strictly separable in such a complex game, the price-prediction component of a trading strategy may be easier to isolate than most. In particular, the problem can be formulated in its own terms, with natural absolute accuracy measures. And in fact, most agent developers choose to define price prediction as a distinct task in their own agent designs.

We divide price prediction into two phases: *initial* and *interim*. Initial refers to the beginning of the game, before any hotel auctions close or provide quote information. Interim refers to the method employed thereafter. Since the information available for initial prediction—flight prices and client preferences—is a strict subset of that available for interim (which adds transaction and hotel price data), most agents treat initial prediction as just a (simpler) special case.

Initial prediction is relevant to bidding policy for hotels, and especially salient for trip choices as these are typically made early in the game. Interim prediction supports ongoing revision of bids as the hotel auctions start to close. As do our price predictors in SAA, Walverine focuses on initial prediction, making minimal adjustments based on current price quotes.[15]

4.9 Walrasian Price Equilibrium in the TAC Market

We now describe how we adapt the Walrasian price equilibrium approach to price prediction for the TAC market. Section 4.2 describes this equilibrium notion in general and Section 4.6 presents our application of it to SAA. Walverine derives its name from its approach to price prediction in TAC: it presumes that TAC markets are well-approximated by a competitive economy and computes a Walrasian competitive price equilibrium as its initial prediction for final hotel prices, taking into account the exogenously determined flight prices.

Let $\boldsymbol{p}$ be the vector of hotel prices, consisting of elements $p_{h,i}$ denoting the price of hotel type h on day i. Let $x^j_{h,i}(\boldsymbol{p})$ denote agent j's demand for hotel h on day i at these prices. Aggregate demand is defined by Equation 4.1. Prices $\boldsymbol{p}$ constitute a Walrasian equilibrium if aggregate demand

[13] The name derives from the seminal economist, Léon Walras, and the University of Michigan mascot, the wolverine.

[14] Indeed, if competitions such as TAC are to be successful in facilitating research, it is necessary to separately evaluate techniques developed for problem subtasks [Stone, 2002].

[15] The nature of these adjustments in TAC [Cheng *et al.*, 2005] is rather different than in SAA given the sequential hotel closings.

equals aggregate supply for all hotels. Since there are 16 rooms available for each hotel on each day, we have that in Walrasian equilibrium, $\boldsymbol{x}(\boldsymbol{p}) = \mathbf{16}$.

Walverine computes an expected demand price equilibrium (EDPE) as described in Section 4.6, estimating $\boldsymbol{x}(\boldsymbol{p})$ as the sum of (1) its own demand based on the eight clients it knows about, and (2) an estimate of expected demand for the other agents (56 clients), based on the specified TAC distribution of client preferences, along with basic assumptions of competitive behavior.

Although equilibrium prices are not guaranteed to exist in TAC, we have found that tatonnement typically produces an approximate equilibrium well within the 300 iterations Walverine devotes to the prediction calculation.

Calculating Expected Demand

A central part of the tatonnement update (Equation 4.2) is determination of demand as a function of prices. This is straightforward if client preferences are known,[16] as it corresponds to an optimization routine that we call the *best-package query* [Cheng *et al.*, 2005] and that addresses what Boyan and Greenwald [2001] call the *completion problem*. But whereas we do know the preferences of our own eight clients, we have no direct knowledge about the 56 clients assigned to the other seven agents.

Therefore, we partition the demand problem into a component from Walverine ($\boldsymbol{x}_w$), and the *expected* demand from the other agents:

$$\boldsymbol{x}(\boldsymbol{p}) = \boldsymbol{x}_w(\boldsymbol{p}) + E[\boldsymbol{x}_{\bar{w}}(\boldsymbol{p})].$$

We calculate $\boldsymbol{x}_w(\boldsymbol{p})$ using a simplified version of Walverine's best-package query (ignoring entertainment holdings) [Cheng *et al.*, 2005]. To calculate $E[\boldsymbol{x}_{\bar{w}}(\boldsymbol{p})]$, we use our knowledge of the type distribution—the distribution from which client preferences are drawn. If agent demand is separable by client,

$$E\left[\boldsymbol{x}_{\bar{w}}(\boldsymbol{p})\right] = E\left[\sum_{i=1}^{56} \boldsymbol{x}_{client_i}(\boldsymbol{p})\right]. \tag{4.8}$$

Since client preferences are i.i.d.,

$$E\left[\boldsymbol{x}_{\bar{w}}(\boldsymbol{p})\right] = 56 \cdot E\left[\boldsymbol{x}_{client}(\boldsymbol{p})\right].$$

At the beginning of the game when there are no holdings of flights and hotels, the agent optimization problem is indeed separable by client, and so (4.8) is justified. At interim points when agents hold goods, the demand optimization problem is no longer separable. However, since we are ignorant about the holdings of other agents, we have no particular basis on which to determine *how* (4.8) is violated, and so we adopt it as an approximation.

It remains to derive a value for $E[\boldsymbol{x}_{client}(\boldsymbol{p})]$. Our solution follows directly from the distribution of clients. Preferred arrival and departure days (pa, pd) are drawn uniformly from the ten possible arrival/departure pairs:

$$E\left[\boldsymbol{x}_{client}(\boldsymbol{p})\right] = \frac{1}{10} \sum_{(pa,pd)} E\left[\boldsymbol{x}_{pa,pd}(\boldsymbol{p})\right]. \tag{4.9}$$

[16] Indeed, we originally validated our prediction concept by applying it to data from the TAC-01 finals, taking the client preferences as given.

For a *given* (pa, pd), the only remaining uncertainty surrounds the hotel premium hp. We observe that the optimal choice of travel days is independent of hp, conditional on hotel choice. Given its hotel type, the hotel premium received by the client is either constant (for T) or zero (for S), regardless of the specific days of stay.

Let $r^*(pa, pd, h)$ denote the optimal trip for the specified day preferences, conditional on staying in hotel h (T or S). We can calculate this trip by taking into account the flight prices, prices for hotel h, day deviation penalties, and expected entertainment bonus (see Section 4.9). Note that the optimal trip for preferences (pa, pd) must be either $r^*(pa, pd, \mathrm{T})$ or $r^*(pa, pd, \mathrm{S})$. Let σ_h denote the net valuation of $r^*(pa, pd, h)$, based on the factors above but *not* accounting for hp.

Hotel premiums are also drawn uniformly, $hp \sim U[50, 150]$. Since T and S differ only in the bonus hp, we can determine the choice based on the relation of σ_{T} and σ_{S}:

$$h = \begin{cases} \mathrm{S} & \text{if } \sigma_{\mathrm{S}} - \sigma_{\mathrm{T}} \geq 150 \\ \mathrm{T} & \text{if } \sigma_{\mathrm{S}} - \sigma_{\mathrm{T}} \leq 50. \end{cases}$$

If instead $50 < \sigma_{\mathrm{S}} - \sigma_{\mathrm{T}} < 150$, then the choice of hotel depends on the actual hp. The uniform distribution of hp entails that the *probability* of S being the optimal choice is

$$\Pr(h = \mathrm{S}) = \frac{\sigma_{\mathrm{S}} - \sigma_{\mathrm{T}} - 50}{100}.$$

Given the choice of trip days and hotel, the demand for this case is established. We aggregate these cases (weighting by probability of hotel choice if applicable) using (4.9), yielding the overall demand per client. Multiplying by 56 gives us $E[\boldsymbol{x}_{\bar{w}}(\boldsymbol{p})]$, and combining with our own demand, finally, the overall expected demand estimate.

Expected Entertainment Surplus

The derivation above deferred detailed explication of our accounting for entertainment bonuses in evaluating alternative trips. We employ estimates of net entertainment contribution as a function of arrival and departure days. Our analysis is based on the distribution of client entertainment preferences, along with the empirical observation (reported by the LivingAgents team [Fritschi and Dorer, 2002]) that entertainment tickets tend to trade at a price near 80. We verified that this indeed obtained during the TAC-01 finals, and refined the estimate by distinguishing the entertainment tickets on congested days 2 and 3 (average price 85.49), from tickets on less congested days 1 and 4 (average price 76.35). Our analysis proceeds by assuming that agents can buy or sell any desired quantity at these prices.

Consider a client staying for d days, with given entertainment values. Its maximal entertainment surplus would be obtained by allocating its most valuable ticket to the cheapest day of its trip if profitable (that is, if the entertainment value exceeds the average price for that day), then if $d \geq 2$, its second most valuable to the next cheapest day, and finally, if $d \geq 3$, its least valuable to a remaining day.

Let x_i denote the cost of the ith least expensive day for the given trip with $1 \leq i \leq \min(3, d)$. The expected entertainment surplus of the trip, then, is given by

$$\sum_{i=1}^{\min(3,d)} EV_i(x_i), \tag{4.10}$$

	Expected Entertainment Surplus	
Arrive:Depart	TAC-01 Prices	TAC-02 Prices
1:2, 4:5	74.7	78.1
1:3, 3:5	101.5	112.2
1:4, 2:5	106.9	120.1
1:5	112.7	121.0
2:3, 3:4	66.2	76.8
2:4	93.0	110.9

Table 4.3: Expected contributions from entertainment, based on prices from TAC-01 and TAC-02 finals, respectively. Walverine employs these summary values in its demand calculations for tatonnement.

where $EV_i(x)$ denotes the expected value of allocating the ith most valuable ticket to a day costing x. Three ticket values are drawn independently from a uniform distribution. Given m i.i.d. draws $\sim U[a,b]$, the ith greatest is less than z with probability

$$F_{i,m}^{[a,b]}(z) = \sum_{j=0}^{i-1} \binom{m}{j} \left(\frac{z-a}{b-a}\right)^{m-j} \left(\frac{b-z}{b-a}\right)^{j}.$$

The expectation of this ith order statistic $Z_{i,m}^{[a,b]}$ is given by

$$E\left[Z_{i,m}^{[a,b]}\right] = a + (b-a)\left(1 - \frac{i}{m+1}\right).$$

We need to determine the expected value of the ith ticket, net of its cost x. The expected surplus of the ith order statistic with respect to x, given that x is between the jth and $(j+1)$st order statistic $(i \leq j)$ is

$$E\left[Z_{i,m}^{[a,b]} - x \;\middle|\; Z_{j+1,m}^{[a,b]} \leq x < Z_{j,m}^{[a,b]}\right] = E\left[Z_{i,j}^{[x,b]} - x\right]. \quad (4.11)$$

The probability of the condition in (4.11) is

$$\Pr(Z_{j+1,m}^{[a,b]} \leq x < Z_{j,m}^{[a,b]}) = F_{j+1,m}^{[a,b]}(x) - F_{j,m}^{[a,b]}(x).$$

Using these expressions, we can sum over the possible positions of x with respect to the order statistics (positions in which value minus cost is positive) to find the expected value of allocating the ith best ticket to a day costing x:

$$\begin{aligned} EV_i(x) &= E\left[\max\left(0, Z_{i,3}^{[0,200]} - x\right)\right] \\ &= \sum_{j=i}^{3} \left(E\left[Z_{i,j}^{[x,200]}\right] - x\right) \left(F_{j+1,3}^{[0,200]}(x) - F_{j,3}^{[0,200]}(x)\right), \end{aligned}$$

providing the value we need to evaluate the expected entertainment surplus for a trip (4.10).

Table 4.3 gives the results of these calculations for each of the ten possible trips.

Interim Price Prediction

The description above covers Walverine's procedure for initial price prediction. Once the game is underway, there are several additional factors to consider.

- Agents already hold flight and hotel goods.
- Flight prices have changed.
- Hotel auctions have issued price quotes, providing a source of information about actual demand.
- Some hotel auctions are closed, precluding further acquisition of these rooms.

Walverine adopts a fairly minimal adjustment of its basic (initial) price prediction method to address these factors. It continues to employ initial flight prices in best-trip calculations, and ignores its own flight and hotel holdings in calculating own demand for open hotels. For closed hotels, it does fix its demand at actual holdings. Since we do not know the holdings of other agents, we make no attempt to account for this in estimating their demand. This applies even to closed hotels—in the absence of information about their allocation, Walverine's tatonnement calculations attempt to balance supply and demand for these as well.

Given price quotes, we modify the price-adjustment process to employ ASK (or final price of closed auctions) as a lower-bound price for each hotel. This constraint is enforced within each iteration of the tatonnement update (4.2).

Distribution-Based Prediction: Price Hedging

Walverine's Walrasian equilibrium analysis results in a point price prediction for each hotel auction. As for SAA (Section 4.5), we expect a distribution-based prediction to improve performance. Some agents, such as ATTac [Stone *et al.*, 2003] and RoxyBot [Greenwald and Boyan, 2004], explicitly generate and use predictions in the form of distributions over prices. Others, including Walverine, generate point predictions but then make decisions with respect to distributions around those estimates. This constitutes a preliminary, ad hoc approach to distribution-based prediction that predates the more sophisticated approach we have applied to SAA. (It would be interesting to apply self-confirming distribution prediction ideas to TAC in future work.)

The greatest source of risk in TAC stems from the possibility that a hotel's price might greatly exceed the estimate, causing the agent to pay a painfully high price or fail to obtain its room(s). Thus, Walverine assigns a small *outlier probability*, π, to the event that a given hotel will reach an unanticipated high price. In the event the hotel is an outlier, we take its price to be $\max(2\hat{p}, 400)$, where $\hat{p}$ is the estimated price of the hotel if it is not an outlier (i.e., according to the equilibrium price-prediction procedure, described above). Walverine's overall price distribution is thus defined by a set of disjoint events with exactly one outlier, at probability π for each of the open hotel auctions, and the residual probability for the event of no outliers.

We apply this price distribution model in our initial calculation of optimal trips, on which we base our starting flight purchases. The resulting choice *hedges* for the potential that some price will deviate significantly from our baseline prediction. The typical effect of our hedging method is to reduce the duration of some trips, thus protecting Walverine somewhat from the exposure problem and to hotel price risk.

4.10 Bidding Strategy using Price Predictions in TAC

At any point in the game, as in SAA, price predictions inform bidding strategy by allowing the predicting agent to better assess bundles on which to bid. In SAA the rest of the bidding strategy is simple: bid the ask price on goods in the preferred bundle that it is not already winning. In TAC, where each minute corresponds to a round of hotel bidding,[17] any of the auctions could be the one to close at the conclusion of any round. Thus, an agent needs to place a serious bid in every hotel auction. To do this, Walverine estimates its *marginal values* for each good using its predicted prices. Let $v^*(g, x)$ denote the value of the best bundle of travel resources, assuming we hold x *additional* units of good g, and taking the price of additional units of g to be infinite. The marginal value of the kth unit of g is $v^*(g, k) - v^*(g, k-1)$. Note that in the case of perfect complements, the marginal value of each good is the full value of the bundle. We have documented our marginal value and preferred bundle calculations [Cheng *et al.*, 2005], but here we take for granted that Walverine has made these estimates.

An agent behaving competitively would bid in hotel auctions by offering to buy units at their marginal values. Again assuming separable clients, this means that each agent will submit an offer for a unit at marginal value for each hotel and each client. Under price uncertainty, the bidding decision is more complicated [Greenwald and Boyan, 2004], but a competitive agent would still not take into account its own effect on prices.

Walverine assumes that other agents bid competitively and itself bids strategically by calculating a set of bids designed to take into account its own effect on hotel prices. This amounts to placing bids that maximize our expected surplus given a distribution from which other bids in the auction are drawn.

Generating Bid Distributions

As for our price-prediction algorithm, we model the seven other agents as 56 individual clients, again assuming zero holdings. Our approach is to generate a distribution of marginal valuations assuming each of the (pa, pd) pairs, and sum over the ten cases to generate an overall distribution f for the representative client.

For a given (pa, pd) pair, we estimate the value of a given room (h, i) as the difference in expected net valuation between the best trip assuming room (h, i) is free, and the best trip of the alternative hotel type h'. In other words, the value of a given room is estimated to be the price above which the client would prefer to switch to their best trip using the alternate hotel type.

Setting the price of (h, i) to zero and that of all other hotels to predicted prices, we calculate best travel bundles $r^*(pa, pd, h)$, $r^*(pa, pd, h')$ and their associated net valuations σ_h and $\sigma_{h'}$ as in Section 4.9. If $(h, i) \notin r^*(pa, pd, h)$, we say that f_h is zero,[18] otherwise it is the expected difference in net valuations:

$$\begin{aligned} f_S &= \max(0, \sigma_S - \sigma_T - hp), \\ f_T &= \max(0, \sigma_T - \sigma_S + hp). \end{aligned}$$

[17] Agents never submit meaningful bids until the end of each minute when another hotel auction is about to close. In fact, in the first year of the competition, before the random hotel closing rule was added, agents did not submit meaningful hotel bids throughout the whole game, until the very end when the most possible information was available, in particular, flight costs. Even though there was an activity rule (auctions with no bids close early) and a nontrivial bid increment, all the bids but the last were just placeholders to keep the auctions open [Stone and Greenwald, 2005].

[18] As for σ, we omit the arguments for pa, pd, and day i where these are apparent from context.

Since $hp \sim U[50, 150]$, these expressions represent uniform random variables:

$$\begin{aligned} \sigma_S - \sigma_T - hp &\sim U[\sigma_S - \sigma_T - 150, \sigma_S - \sigma_T - 50], \\ \sigma_T - \sigma_S + hp &\sim U[\sigma_T - \sigma_S + 50, \sigma_T - \sigma_S + 150]. \end{aligned} \quad (4.12)$$

For each (pa, pd) we can thus construct a cumulative distribution $F_{pa,pd}$ representing the marginal valuation of a given hotel room. In general, $f_{pa,pd}$ will include a mass at zero, representing the case where the room is not used even if free. Thus, we have

$$F_{pa,pd}(x) = \begin{cases} 0 & \text{if } x < \max(0, \alpha) \\ \dfrac{x - \alpha}{\beta - \alpha} & \text{if } \max(0, \alpha) \le x \le \beta \\ 1 & \text{if } x \ge \beta \end{cases}$$

where α and β are the lower and upper bounds, respectively, of the corresponding uniform distribution of (4.12).

The overall valuation distribution for a representative client is the sum over arrival/departure preferences,

$$F(x) = \frac{1}{10} \sum_{(pa,pd)} F_{pa,pd}(x).$$

Finally, it will also prove useful to define a valuation distribution conditional on exceeding a given value q. For $x \ge q$,

$$F(x \mid q) = \frac{F(x) - F(q)}{1 - F(q)}. \quad (4.13)$$

Computing Optimal Bids

After estimating a bid distribution, Walverine derives an optimal set of bids with respect to this distribution. Our calculation makes use of an order statistic, $F_{k,n}(x)$, which represents the probability that a given value x would be kth highest if inserted into a set of n independent draws from F.

$$F_{k,n}(x) = [1 - F(x)]^{k-1} \, F(x)^{n-k+1} \binom{n}{k-1}$$

We can also define the conditional order statistic, $F_{k,n}(x \mid q)$, by substituting the conditional valuation distribution (4.13) for F in the definition above.

Once hotel auctions start issuing price quotes, we have additional information about the distribution of bids. If H is the *hypothetical quantity won*[19] for Walverine at the time of the last issued quote, the current ASK tells us that there are $16 - H$ bids from other clients at or above ASK, and $56 - (16 - H) = 40 + H$ at or below (assuming a bid from every client, including zero bids). We therefore define another order statistic, B_k, corresponding to the kth highest bid, sampling $16 - H$ bids from $F(\cdot \mid ASK)$ as defined by (4.13), and $40 + H$ bids from F.

Note that our order statistics are defined in terms of other agents' bids, but we are generally interested in the kth highest value in an auction overall. Let n_b be the number of our bids in the

[19] This value, indicating how many units the agent would win if the auction closed immediately, is available from the auction as part of the quote information.

auction greater than b. We define B_k so as to include our own bids, and employ the $(k - n_b)$th order statistic on others, $F_{k-n_b,n}(b)$, in calculating B_k.

Given our definitions, the probability that a bid b will be the kth highest is the following:

$$B_k(b) = \sum_{i=0}^{k-n_b-1} F_{i,16-H}(b) \cdot F_{k-n_b-i,40+H}(b \mid ASK). \tag{4.14}$$

We characterize the expected value of submitting a bid at price b as a combination of the following statistics, all defined in terms of B_k.

- $B_{16}(b)$: Probability that b will win and set the price.
- $B_{15}^{+} \equiv \sum_{i=1}^{15} B_i(b)$: Probability that b will win but not set the price
- $M_{15} \equiv \{x \mid \sum_{i=1}^{15} P_i(x) = .5\}$: Median price if we submit an offer b.
- $M_{16} \equiv \{x \mid \sum_{i=1}^{16} P_i(x) = .5\}$: Median price if we do not bid.

Before proceeding, we assess the quality of our model, by computing the probability that the 16th bid would be above the quote given our distributions. If this probability is sufficiently low: ($B_{16}^{+}(ASK) < .02$) then we deem our model of other agents' bidding to be invalid and we revert to our most conservative bid: our own marginal value.

If the conditional bid distribution passes our test, based on these statistics we can evaluate the expected utility EU of a candidate bid for a given unit, taking into consideration the marginal value MV of the unit to Walverine, and the number of units n_b of this good for which we have bids greater than b. Expected utility of a bid also reflects the expected price that will be paid for the unit, as well as the expected effect the bid will have on the price paid for all our higher bids in this auction. Lacking an expression for expected prices conditional on bidding, we employ as an approximation the median price statistics, M_{15} and M_{16}, defined above.[20]

$$\begin{aligned} EU(b) \quad = \quad & B_{16}(b)\,[(MV - b) - n_b(b - M_{16})] \\ & + B_{15}^{+}(b)\,[(MV - M_{15}) - n_b(M_{15} - M_{16})] \end{aligned}$$

Walverine's proposed offer for this unit is the bid value maximizing expected utility,

$$b^* = \arg\max_b EU(b) \tag{4.15}$$

which we calculate by brute force evaluation of $EU(b)$.

Upon calculating desired offer prices for all units of a given hotel, Walverine assembles them into an overall bid vector for the auction, subject to additional adjustments for the beat-the-quote bidding rule. We describe these adjustments, along with analysis of our bidding strategy in our full description of Walverine [Cheng *et al.*, 2005].

4.11 Evaluating Prediction Quality in TAC

Having described Walverine's price prediction method and concomitant bidding strategy, we now consider two ways of assessing prediction quality. The first is based purely on how well predicted prices $\hat{\boldsymbol{p}}$ match realized prices $\boldsymbol{p}$. The second measures prediction quality in terms of resulting performance in the game.

[20] Offline analysis using Monte Carlo simulation verified that the approximation is reasonable.

Euclidean Distance

Perhaps the most straightforward measure of the closeness of two price vectors is their Euclidean distance:

$$d(\hat{\boldsymbol{p}}, \boldsymbol{p}) \equiv \left[\sum_{(h,i)} (\hat{p}_{h,i} - p_{h,i})^2 \right]^{1/2},$$

where (h, i) indexes the price of hotel $h \in \{\mathrm{S}, \mathrm{T}\}$ on day $i \in \{1, 2, 3, 4\}$. Clearly, lower values of d are preferred, and for any $\boldsymbol{p}$, $d(\boldsymbol{p}, \boldsymbol{p}) = 0$.

Note that if $\hat{\boldsymbol{p}}$ is in the form of a distribution, the Euclidean distance of the mean provides a lower bound on the average distance of the components of this distribution. Thus, at least according to this measure, evaluating distribution predictions in terms of their means provides a bias in their favor.

It is likewise the case that among all constant predictions, the actual mean $\bar{\boldsymbol{p}}$ for a set of games minimizes the aggregate *squared* distance for those games. That is, if $\boldsymbol{p}^j$ is the actual price vector for game j, $1 \leq j \leq N$,

$$\bar{\boldsymbol{p}} \equiv \frac{1}{N} \sum_{j=1}^{N} \boldsymbol{p}^j = \arg\min_{\hat{\boldsymbol{p}}} \sum_{j=1}^{N} [d(\hat{\boldsymbol{p}}, \boldsymbol{p}^j)]^2.$$

There is no closed form for the prediction minimizing aggregate d, but one can derive it numerically for a given set of games [Bose *et al.*, 2003].

Expected Value of Perfect Prediction

Euclidean distance d appears to be a reasonable measure of accuracy in an absolute sense. However, the purpose of prediction is not accuracy for its own sake, but rather to support decisions based on these predictions. Thus, we seek a measure that relates in principle to expected TAC performance. By analogy with standard value-of-information measures, we introduce the concept of *value of perfect prediction* (*VPP*).

Suppose an agent could anticipate perfectly the eventual closing price of all hotels. Then, among other things, the agent would be able to purchase all flights immediately with confidence that it had selected optimal trips for all clients.[21] Since most agents commit to (at least some) trips at the beginning of the game, perfect prediction would translate directly to improved quality of these choices. We take this ability to choose optimal trips as the primary worth of predictions, and measure quality of a prediction in terms of how it supports trip choice in comparison with perfect anticipation. The idea is that VPP will be particularly high for agents that otherwise have a poor estimate of prices. If we are already predicting well, then the value of obtaining a perfect prediction will be relatively small. This corresponds to the use of standard value-of-information concepts for measuring uncertainty: for an agent with perfect knowledge, the value of additional information is nil.

Specifically, consider a client c with preferences (pa, pd, hp). A trip's surplus for client c at prices $\boldsymbol{p}$, $\sigma_c(r, \boldsymbol{p})$ is defined as for a bundle of goods in SAA (Equation 4.3):

$$\sigma_c(r, \boldsymbol{p}) \equiv v_c(r) - \mathit{cost}(r, \boldsymbol{p}),$$

[21] Modulo some residual uncertainty regarding availability of entertainment tickets, which we ignore in this analysis.

where *cost*$(r, \boldsymbol{p})$ is simply the total price of flights and hotel rooms included in trip r. Let

$$r_c^*(\boldsymbol{p}) \equiv \arg\max_r \sigma_c(r, \boldsymbol{p})$$

denote the trip that maximizes surplus for c with respect to prices $\boldsymbol{p}$. Given these definitions, the expression

$$\sigma_c(r_c^*(\hat{\boldsymbol{p}}), \boldsymbol{p})$$

represents the surplus of the trip *chosen based on* prices $\hat{\boldsymbol{p}}$, but *evaluated with respect to* prices $\boldsymbol{p}$. From this we can define value of perfect prediction,

$$\mathit{VPP}_c(\hat{\boldsymbol{p}}, \boldsymbol{p}) \equiv \sigma_c(r_c^*(\boldsymbol{p}), \boldsymbol{p}) - \sigma_c(r_c^*(\hat{\boldsymbol{p}}), \boldsymbol{p}).$$

Note that our *VPP* definition is relative to client preferences, whereas we seek a measure applicable to a pair of price vectors outside the context of a particular client. To this end we define the *expected value of perfect prediction*, *EVPP*, as the expectation of *VPP* with respect to TAC's distribution of client preferences:

$$\begin{aligned} \mathit{EVPP}(\hat{\boldsymbol{p}}, \boldsymbol{p}) &\equiv E_c[\mathit{VPP}_c(\hat{\boldsymbol{p}}, \boldsymbol{p})] \\ &= E_c[\sigma_c(r_c^*(\boldsymbol{p}), \boldsymbol{p})] - E_c[\sigma_c(r_c^*(\hat{\boldsymbol{p}}), \boldsymbol{p})]. \end{aligned} \tag{4.16}$$

As for d, lower values of *EVPP* are preferred, and for any $\boldsymbol{p}$, *EVPP*$(\boldsymbol{p}, \boldsymbol{p}) = 0$. From (4.16) we see that computing *EVPP* reduces to computing $E_c[\sigma_c(r_c^*(\hat{\boldsymbol{p}}), \boldsymbol{p})]$. We can derive this latter value as follows. For each (pa,pd) pair, determine the best trip for hotel S and the best trip for hotel T, respectively, at prices $\hat{\boldsymbol{p}}$ ignoring any contribution from hotel premium, hp. From this we can determine the threshold value of hp (if any) at which the agent would switch from S to T. We then use that boundary to split the integration of surplus (based on prices $\boldsymbol{p}$) for these trips, with respect to the underlying distribution of hp. This procedure is analogous to Walverine's method for calculating expected client demand (Section 4.9) in its Walrasian equilibrium computation.

Results from TAC-02

After the 2002 competition we completed an extensive survey [Wellman *et al.*, 2004] of agent strategies, including prediction methods, information employed for predictions, and relative performance using our two accuracy measures described above. Figure 4.2 plots agent performance on both measures—d and *EVPP*—for thirteen TAC agents for which we were able to obtain prediction data. We conclude that, up to statistical ties, no prediction strategy (significantly) outperformed Walverine's Walrasian equilibrium approach by either measure. Furthermore, incorporating flight prices to predict hotel prices is the key distinguishing factor for prediction accuracy. It is possible to employ machine learning, as one agent (ATTac) did, to induce a function from flight prices to hotel prices based on historical data. Walverine was able to achieve equal accuracy using its analytic Walrasian model, using no historical data. We additionally find [Wellman *et al.*, 2004] that incorporation of own client data has only a minimal effect on prediction accuracy, by either measure. We expect this will hold in SAA as well, where we do not currently employ own type for prediction.

4.12 Conclusion

Although TAC is a far more complex game than SAA, they share a common core: simultaneous ascending auctions for complementary goods. I have described in this chapter common prediction-

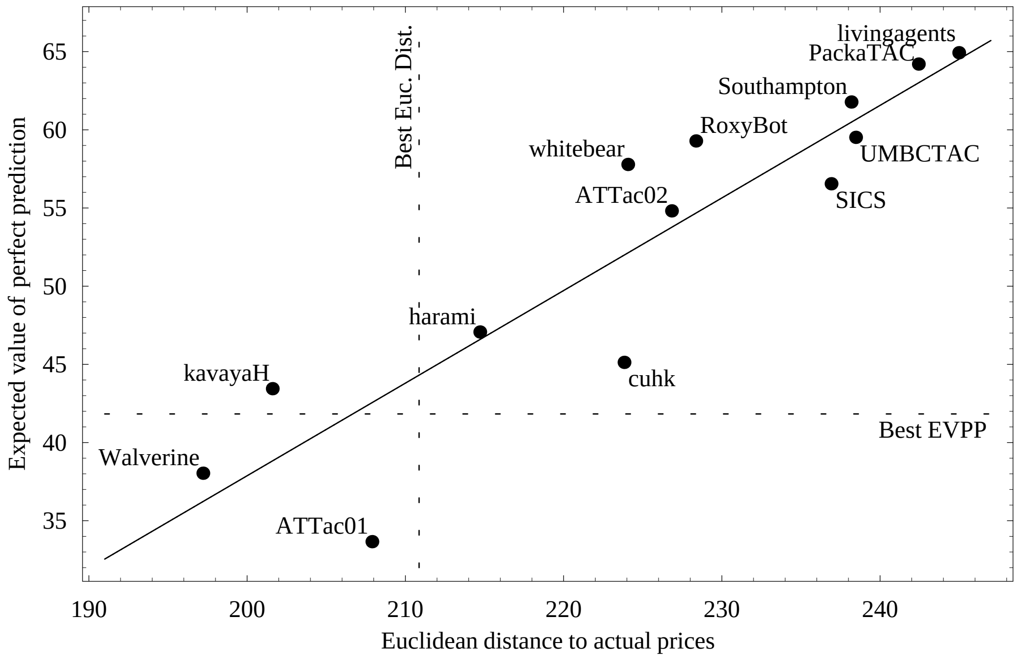

Figure 4.2: Prediction quality for thirteen TAC-02 agents. Dashed lines delimit the accuracy achievable with constant predictions (independent of flight prices and own client preferences): "best Euclidean distance" and "best EVPP" for the two respective measures. The diagonal line is a least-squares fit to the points. Observe that the origin of this graph is at (190,32).

based approaches for both domains, forming the basis for a range of strategies which we analyze in the remaining chapters, first for SAA and then for TAC.

Focusing on the prediction subtask, our analysis also introduces a new measure, expected value of perfect prediction, which captures the instrumental value of accurate predictions, beyond their nominal correspondence to realized outcomes. We believe it striking that Walverine's purely analytic approach, without any empirical tuning, could achieve accuracy comparable—both with the absolute measure and the performance-based measure—to the best available machine-learning method (ATTac-01). Moreover, many would surely have been skeptical that straight competitive analysis could prove so successful, given the manifest unreality of its assumptions as applied to TAC. Our analysis does not show that Walrasian equilibrium is the best possible model for price formation in TAC, but it does demonstrate that deriving the shape of a market from an idealized economic theory can be surprisingly effective.

For SAA, self-confirming prediction strategies provide a definitive answer to the question of absolute prediction accuracy. In the next chapter we demonstrate game-theoretically that playing a self-confirming distribution prediction strategy is a robust strategy in SAA.

Chapter 5

Empirical Game Analysis for Simultaneous Ascending Auctions

IN WHICH *we apply our empirical game methodology to the analysis of promising strategies in restricted SAA games and conclude that self-confirming distribution prediction is hard to beat (by much) in a range of environments.*

The previous chapter presents a family of bidding strategies for SAA called *perceived price bidders*. These include straightforward bidding (SB), sunk-aware strategies, and most notably, price predicting strategies. Several methods for constructing point price predictions and price distributions are presented: Walrasian price equilibrium, empirical distributions from historical data, and self-confirming predictions. In this chapter, we apply the techniques of Chapter 3 to analyze SAA games restricted to subsets of these strategies.

5.1 Type Distributions for SAA

A symmetric game like SAA is defined by its payoff function, the strategy space, and the common-knowledge type distribution. As described in Section 4.3, we study SAA games parameterized by the number of agents, n, the number of goods, m, and the common-knowledge type distribution giving valuation functions over subsets of goods. Here I describe the type distributions employed in the experiments in this chapter.

We consider agent valuation functions of a restricted form in which an agent's value for a bundle depends on getting a minimum number of goods and on the highest numbered good in the bundle given that it hits its minimum. Letting $X_{(i)}$ denote the ith smallest element of $X \subseteq \{1, \ldots, m\}$, agent j's valuation for the bundle is

$$v_j(X) = \begin{cases} 0 & \text{if } |X| < \lambda_j \\ \omega^j_{X_{(\lambda_j)}} & \text{otherwise,} \end{cases}$$

where λ_j is the number of goods agent j desires (it has no value for getting less than λ and no additional value for getting more) and ω^j_i is j's value if i is the λth good in the ordered bundle. In other words, the agent throws away all but the first λ goods (lower numbered goods have more value) and then derives value based only on the highest numbered remaining good.

This model of agent valuation is taken from our work in applying SAA to scheduling [Reeves *et al.*, 2005]. In that interpretation, goods are time slots (lower numbered goods are earlier slots) and an agent needs λ slots to finish its job, deriving value based on the earliest slot by which it can finish. Our valuation model would also be apt for other classes of goods (physical or otherwise), arranged in order of decreasing quality, where an agent needs no more or less than a certain number of goods and its value depends on the lowest quality good in the bundle, not counting excess goods. To coin a canonical example, imagine bidding for individual chain links to build a chain of a certain length. Regardless of its realism or scope of applicability, we employ type distributions restricted in this way because they capture fundamental complementarities in diffuse distributions with a minimum of free parameters.[1] Specifically, the distribution parameters our model affords are

- the desired bundle size, λ,
- the "weakest link" values for the m goods (for each good i, the value of a bundle in which i is the λth good), denoted by the vector of decreasing values, $\boldsymbol{\omega}$, and
- the maximum value an agent can have for a bundle, V.

The primary structural distinction we explore is with respect to the distributions of the desired bundle size, λ.

Uniform In the uniform model λ is a random variable with distribution $\lfloor Z \rfloor$ where $Z \sim U[1, m+1]$. That is, $\Pr(\lambda = x) = \dfrac{1}{m}, \forall x \in \{1, \ldots, m\}$. We denote an SAA game using the uniform model by $\mathrm{SAA}_{\lambda \sim U}$.

Constant In the constant model, λ_j is a fixed constant for all j. In the experiments in this chapter, $\lambda_j = 2$ and we denote this game by $\mathrm{SAA}_{\lambda=2}$.

Exponential In the exponential model, denoted $\mathrm{SAA}_{\lambda \sim E}$, λ is drawn from an exponential distribution:

$$\Pr(\lambda = i) = \begin{cases} 2^{-i} & \text{if } i \in \{1, \ldots, m-1\} \\ 2^{-m+1} & \text{if } i = m. \end{cases}$$

In all three of these models, the vector $\boldsymbol{\omega}$ is initialized with integers $\sim \lfloor U[1, V+1] \rfloor$, but then modified to ensure that they decrease monotonically (since our agent valuations are defined only for this case):[2]

$$\omega_i \leftarrow \omega_{\min\{i' \geq i \mid \omega_{i'} \leq \omega_{i-1}\}} \text{ or } 0, \forall i \in \{\lambda + 1, \ldots, m\}.$$

In words, iterate over $\boldsymbol{\omega}$ and whenever an element violates monotonicity (i.e., exceeds its predecessor) set it to the earliest later value that restores monotonicity (i.e., is less than or equal to its predecessor).

[1] Leyton-Brown *et al.* [2000] have characterized a suite of valuation distributions in the context of combinatorial auction mechanisms that could be profitably employed in future studies.

[2] This method was chosen over the more straightforward $\omega_i \leftarrow \min(\omega_{i-1}, \omega_i), \forall i \in \{\lambda + 1, \ldots, m\}$ as a vestige of an implementation that represented $\boldsymbol{\omega}$ as a sparse vector.

5.2 Experiments in Sunk-Awareness

We begin with a systematic exploration of bidding strategies that vary on the sunk-awareness parameter, k, defined in Section 4.4. We consider parameter settings in multiples of 1/20 from 0 to 1. For simplicity of reference, we designate strategies by an integer from 0 to 20, such that strategy i refers to a sunk-aware agent with $k = i/20$. Thus strategy 20 denotes straightforward bidding (SB; confer Section 4.4). All of our sunk-awareness experiments are for five goods, five agents, and maximum bundle value 50 ($m = n = 5$ and $V = 50$) except in Section 5.2 where we vary the number of agents from two to ten.

Uniform Valuations Model

In Figure 5.1 we offer a representation of the payoff matrix for $\mathrm{SAA}_{\lambda\sim U}$ with strategies 18, 19, and 20. The first column represents the payoffs for the strategy profile $\langle 18, 18, 18, 18, 18\rangle$. Each strategy in this profile receives the same payoff (since each agent is playing the same strategy and the game is symmetric) of about 1.12. The second column presents the payoffs for $\langle 18, 18, 18, 18, 19\rangle$. When playing against one $k = 19/20$ agent, the other four $k = 18/20$ agents now do better than in the all-18 profile, and very slightly better than the sole $k = 19/20$ agent does in this profile. In the all-18 profile, when one agent deviates from 18 to 20 (third column), it does noticeably better and so do the agents playing 18.

In Figure 5.2 we show the result of solving this game using replicator dynamics (Section 3.6). The population evolves to all playing 20. This is in fact a Nash equilibrium as can be seen by noting that the all-20 profile in the payoff matrix (Figure 5.1) scores higher than any unilateral deviation. In this restricted game, 20 is a dominant strategy (this can be verified, albeit tediously, by inspecting the payoff matrix), and hence the only Nash equilibrium.

We have confirmed this result with both GAMBIT (Section 2.1) and amoeba (Section 3.6). We also find that the replicator dynamics converges to the unique Nash equilibrium from various initial population proportions (for example, see Figure 5.3). Recall that strategy 20 corresponds to straightforward bidding (no sunk awareness).

Constant Valuations Model

As we know from Section 4.4, it is not the case that SB is a dominant strategy. Even when we restrict strategies to straightforward bidding extended only by the sunk awareness parameter (k), we can find environments in which $k = 1$ does not dominate.

In our experiments with $\mathrm{SAA}_{\lambda=2}$ we consider a slightly larger set of strategies. In our first run, we consider strategies 16, 17, 18, 19, and 20. We present the replicator dynamics for our estimate of the expected payoff matrix in Figure 5.4. The payoff matrix required 22 million game simulations for each of the 126 strategy profiles. When run through the replicator dynamics, the system evolves to $\langle 0.745, 0.255, 0, 0, 0\rangle$ which constitutes a mixed-strategy Nash equilibrium. Convergence to this equilibrium is robust to a variety of different initial population proportions. Note that in this environment, SB is not even supported. Instead, the most sunk-aware (i.e., lowest k) strategies have greatest weight in the mixed strategy. GAMBIT reveals that strategies 19 and 20 are dominated.

We have not verified that this is a unique equilibrium; GAMBIT (which attempts to find all equilibria) was not able to find a symmetric equilibrium for this game after days of cpu time (it did find one asymmetric equilibrium). Amoeba takes about 15 minutes to find this equilibrium.

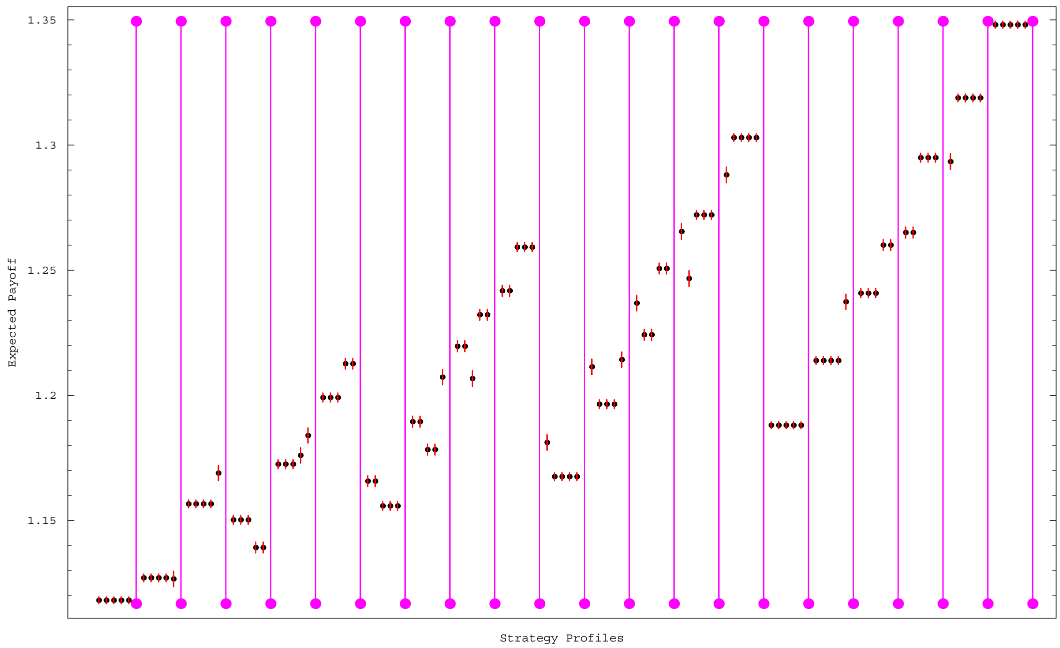

Figure 5.1: Payoff matrix for $\mathrm{SAA}_{\lambda\sim U}$ restricted to strategies $20k \in \{18, 19, 20\}$. Each column corresponds to a strategy profile: $\langle 18, 18, 18, 18, 18\rangle$ through $\langle 20, 20, 20, 20, 20\rangle$ in lexicographic order. The jth dot within a column represents the mean payoff for the jth strategy in the profile. This payoff matrix is based on over 45 million games simulated for each of the 21 profiles, using the brute-force method of Section 3.3 and requiring weeks of cpu time. The error bars denote 95% confidence intervals.

Given that 16 was the most heavily represented strategy when the game is restricted to the 16–20 range, it is natural to investigate whether lower values might perform better still. We tested the above game with a broader but coarser grid of strategies: 0, 8, 12, 16, and 20. We show the evolutionary dynamics in Figure 5.5 based on a payoff matrix estimated by eight million simulations per profile. The evolved Nash equilibrium is for everyone to play 16. According to GAMBIT, strategies 0 and 8 are dominated, and everyone playing 16 is the only Nash equilibrium.

This game stressed two of our solution methods: it took GAMBIT about a day of runtime to reach its conclusion. The amoeba algorithm did not find any Nash equilibria at all (though it identified a mixed strategy close to pure strategy 16 as nearly in equilibrium).

Exponential Valuations Model

Our final variation on the agent valuation model applies the exponential model. We present the evolutionary dynamics for the strategy set 16–20 in Figure 5.6, which is based on a payoff matrix estimated from 22 million samples per profile. The system evolves to $\langle 0, 1, 0, 0, 0\rangle$ (i.e., everyone play 17), which is a Nash equilibrium. This equilibrium is robust to initial population distribution using replicator dynamics. Amoeba does not find this equilibrium, but again identifies a nearby mixed strategy as close. GAMBIT determined that no strategy was dominated in this game, and

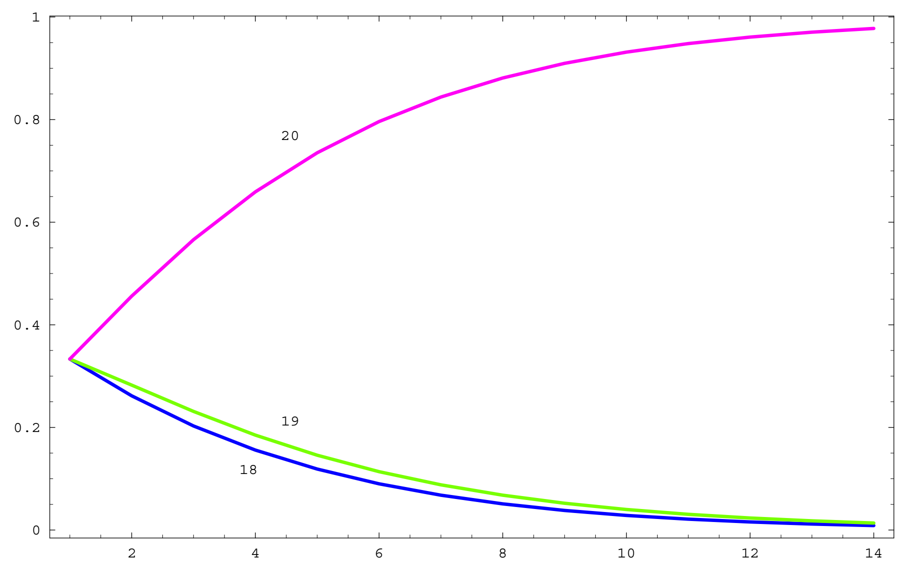

Figure 5.2: First 14 generations of replicator dynamics for strategies $20k \in \{18,\ 19,\ 20\}$ in $\mathrm{SAA}_{\lambda \sim U}$. Strategy 20 quickly takes over the population, hence the evolved equilibrium is $\langle 0, 0, 1 \rangle$ meaning everyone plays 20.

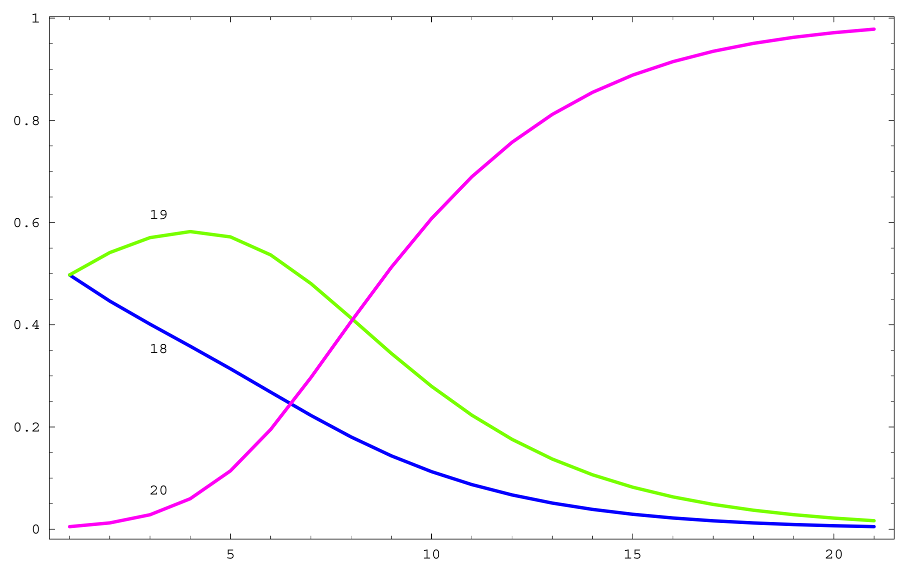

Figure 5.3: Replicator dynamics for the same game as shown in Figure 5.2, but with 100 times fewer 20s in the initial population as 18s or 19s.

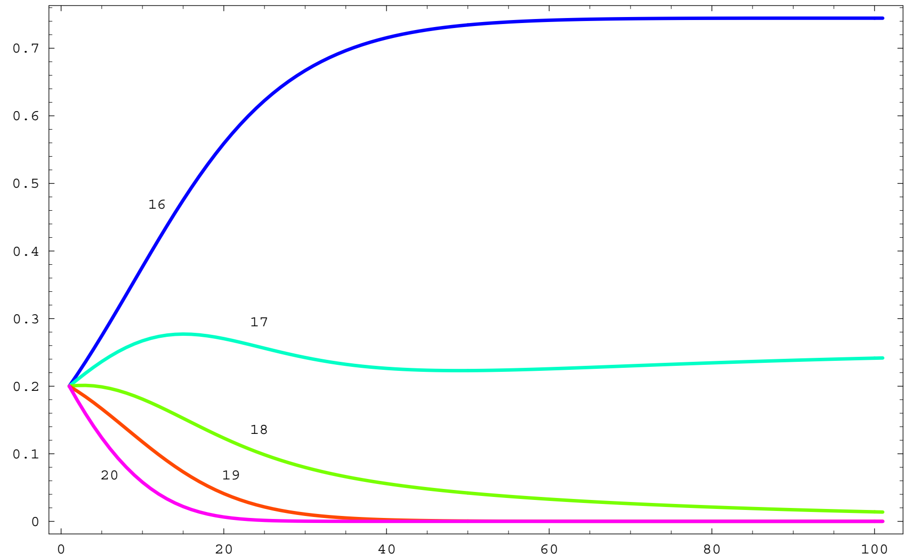

Figure 5.4: Replicator dynamics for strategies $20k \in \{16, \ldots, 20\}$ in $\mathrm{SAA}_{\lambda=2}$. The evolved Nash equilibrium is $\langle 0.745, 0.255, 0, 0, 0 \rangle$.

was unable to determine whether the equilibrium is unique. From this configuration, the exponential model yields an observed equilibrium distribution for k intermediate between the uniform and constant models.

Varying Number of Players

The experiments reported above all employ a five-player game configuration. We have performed further trials varying the number of agents ($n = 2, 8, 10$), maintaining the other game parameters as in our standard setup, under the exponential valuations model (Section 5.2). The objective of this variation was to exercise the methodology on a range of settings of game shapes, and to identify any systematic relation between n and the equilibria we find.

Two Agents

With only two players, we can consider a larger set of candidate strategies. We investigated 14 strategies, defined by the set $20k \in \{0, 3, 6, 8, 10, 11, \ldots, 17, 18, 20\}$. This yields 105 profiles, for each of which we simulated 1.2 million games to construct the payoff matrix, depicted in Figure 5.7. Replicator dynamics (shown in Figure 5.8) finds a Nash equilibrium in which all agents play 15. GAMBIT identifies this as one of three equilibria (all symmetric) for this payoff matrix. All playing 14 is also an equilibrium, as is the mixed strategy of playing 14 with probability 0.514 and 15 otherwise. Only strategies 14, 15, and 16 survive iterated elimination of dominated strategies. We see in Figure 5.8 that these are indeed the three most tenacious strategies under our replication process.

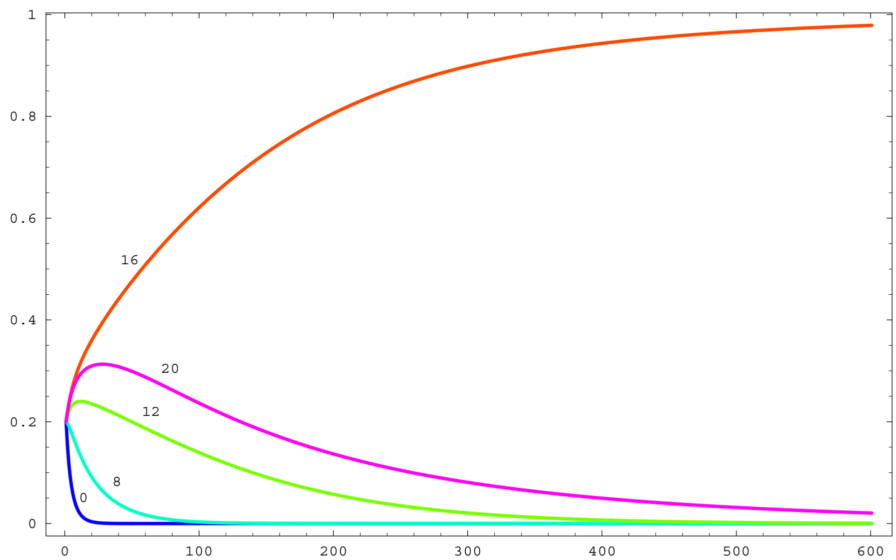

Figure 5.5: Replicator dynamics for strategies $20k \in \{0, 8, 12, 16, 20\}$ in $\mathrm{SAA}_{\lambda=2}$. The evolved equilibrium is $\langle 0, 0, 0, 1, 0 \rangle$—i.e., everyone play 16.

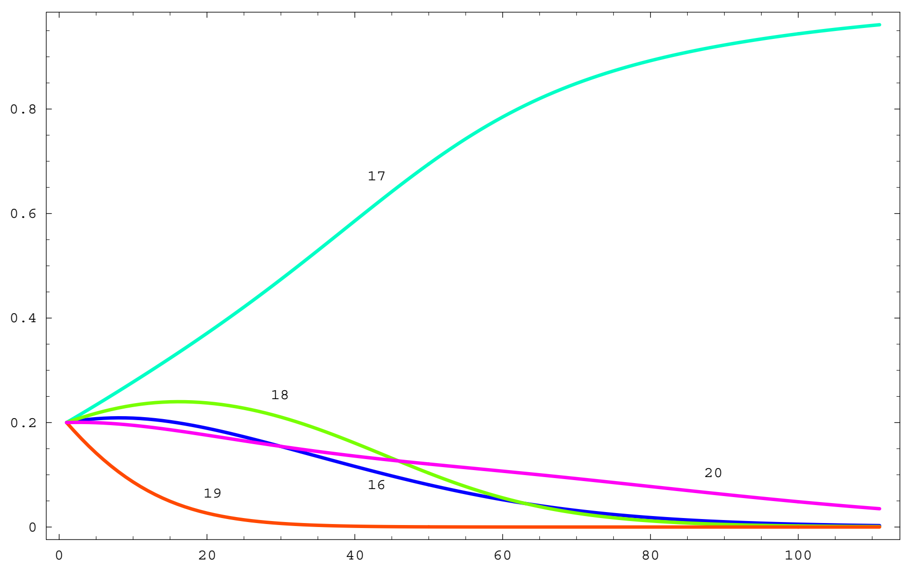

Figure 5.6: Replicator dynamics for strategies $20k \in \{16, \ldots, 20\}$ in $\mathrm{SAA}_{\lambda\sim E}$. The evolved equilibrium is $\langle 0, 1, 0, 0, 0 \rangle$—i.e., everyone play 17.

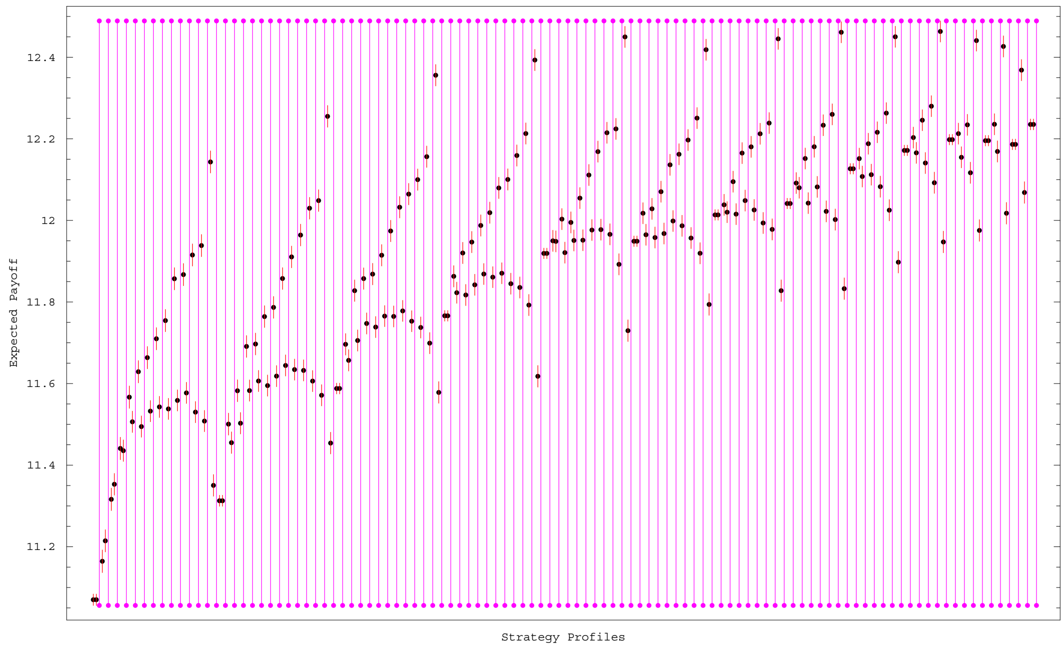

Figure 5.7: Payoff matrix for two-player $\mathrm{SAA}_{\lambda\sim E}$ with strategies $20k \in \{0, 3, 6, 8, 10, 11, \ldots, 17, 18, 20\}$.

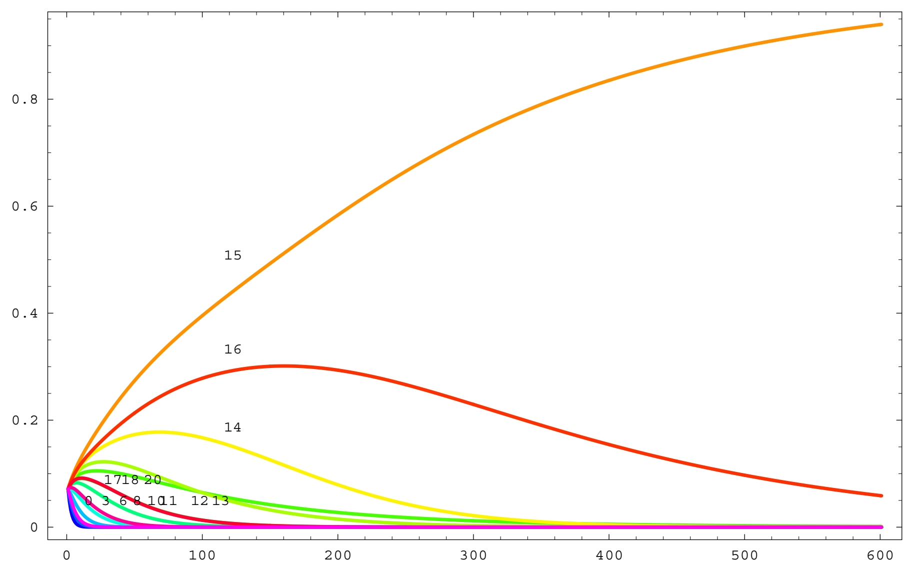

Figure 5.8: Replicator dynamics for two-player $\mathrm{SAA}_{\lambda\sim E}$ with strategies $20k \in \{0, 3, 6, 8, 10, 11, \ldots, 17, 18, 20\}$.

Eight and Ten Agents

In this chapter we do not pursue the variance reduction or player reduction techniques of Chapter 3 (the experiments here predate our discovery/development of those techniques). Thus, with more than a handful of agents, it is not feasible to create a payoff matrix with more than a handful of strategies. Our experiments with eight and ten-player games employ a pool of four strategies: $20k \in \{10, 14, 17, 20\}$. This yields 165 profiles for the eight-player case, for which we simulated 1.5 million games per profile. For the ten-player case there are 286 profiles. We simulated 3.9 million games per profile, which took many cpu weeks.

The conclusion for both eight and ten players is the same: 20 is a dominant strategy. In both cases, the replicator dynamics show strategy 20 overwhelming the population within 40 generations. For the eight-player case, GAMBIT confirms that 20 is dominant (and therefore also the unique Nash equilibrium). But for ten players, the raw payoff matrix (i.e., the normal form without exploiting symmetry) contains 10 million payoff values. GAMBIT does not exploit symmetry, and in our installation, crashes trying to load this game into memory.

Discussion

In our experiments with the exponential valuation model, the equilibrium k value is monotone in the number of agents, n. This can be explained by observing that increasing n can ameliorate the exposure problem. Consider the situation when the prices pass the threshold at which an agent stops bidding. The presence of more competing bidders increases the likelihood that the stopped agent will be let off the hook for its current winnings by being outbid. Therefore, it is less compelling for an agent to treat its current winnings as a sunk cost. In other words, k should be closer to 1 the more agents there are, which is what we find here.

5.3 Sensitivity Analysis for Sunk-Awareness Results

Section 3.8 describes our methods of sensitivity analysis for assessing the quality of equilibria found in empirical payoff matrices. For our sunk-awareness results, we focus on the method of assessing the confidence on mixture probabilities.

We conclude that several of our results reported above are impervious to sampling noise. This was determined by performing our equilibrium analysis on several thousand variants and finding that the equilibrium never changed. This was the case for results reported for $\mathrm{SAA}_{\lambda\sim U}$ in Section 5.2 and for the $\mathrm{SAA}_{\lambda\sim E}$ games with eight and ten players reported in Section 5.2. For our other results we find varying amounts of sensitivity. Figures 5.9, 5.10, 5.11, and 5.12 illustrate this by showing the cumulative distribution functions for the equilibrium proportions of each of the strategies. The dotted vertical lines show the mean proportion for the corresponding strategy over all the variant payoff matrices sampled.

For example, we see in Figure 5.9 that for the $\mathrm{SAA}_{\lambda=2}$ game reported in Section 5.2, the mean proportion of strategy 16 is identical to the proportion found for the maximum likelihood payoff matrix (using the actual sample means) and it varies according to a near-perfect normal distribution. We also see that strategy 18, which died out for the maximum likelihood payoff matrix has a 10% chance of actually holding on to 5% of the population in equilibrium. Note that since this measure is conservative, the true equilibrium results are actually more likely to match those reported in Section 5.2 for the max-likelihood payoff matrices.

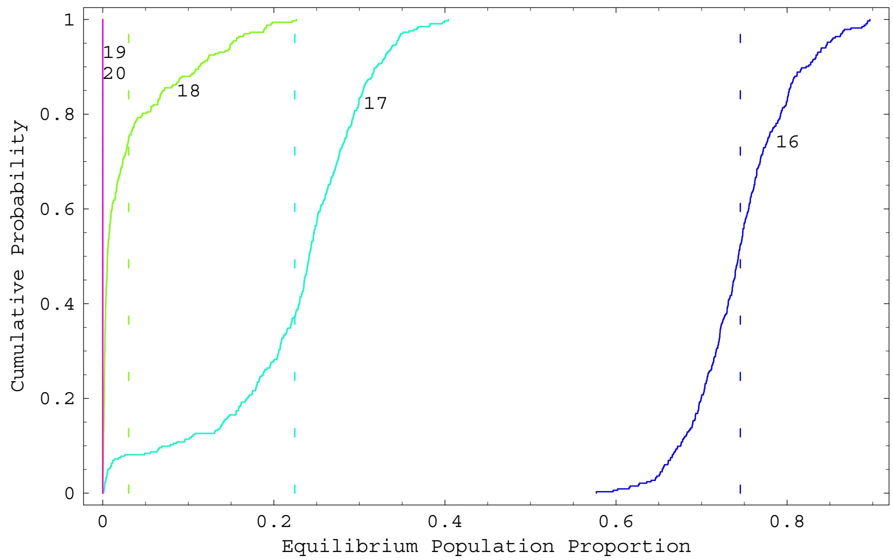

Figure 5.9: Sensitivity analysis for five-player $\mathrm{SAA}_{\lambda=2}$ game with strategies $20k \in \{16, \ldots, 20\}$. Compare to the replicator dynamics for the maximum likelihood payoff matrix for this game in Figure 5.4.

Figure 5.12 indicates that equilibria reported in Section 5.2 may be quite sensitive to sampling noise. However, we have run a smaller two-player experiment with nine strategies ($20k \in \{0, 3, 6, 8, 10, 12, 14, 17, 20\}$) where 14 was dominant and found this result to be perfectly robust to sampling noise. Therefore, the qualitative conclusions about this game are not seriously suspect. Nonetheless, we ran an additional three million simulations per profile and found some slight changes: strategy 16 no longer survives iterated elimination of dominated strategies and the mixed strategy equilibrium is skewed more towards strategy 14. There was no change in the pure strategy equilibria. The sensitivity analysis with the additional games shows greater robustness to sampling error with the additional samples.

5.4 Baseline Price Prediction vs. Straightforward Bidding

We turn now to price prediction strategies for SAA. The first question we address is how our baseline point price prediction strategy ($\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{SB}})$; see Section 4.6) fares against straightforward bidding (SB; see Section 4.4). We find that this unsophisticated prediction method improves performance substantially. We then measure the contribution of two separate aspects of the price prediction strategy to improved performance. We conduct these analyses for $\mathrm{SAA}_{\lambda\sim U}$ with $n = m = 5$ and $V = 50$, employing the computational game-theoretic methodology of Chapter 3.

Table 5.1 on page 89 shows the predicted prices for $\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{SB}})$, i.e., the average prices realized by a profile of all SB agents, based on one million simulated games. To analyze the bidding policy based on this prediction, we consider a game restricted to the two strategies SB and $\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{SB}})$. We again generate the expected payoff matrix for this game using the brute-force method of Section 3.3. The result is depicted in Figure 5.13 (the representation is simpler than for the case of games with

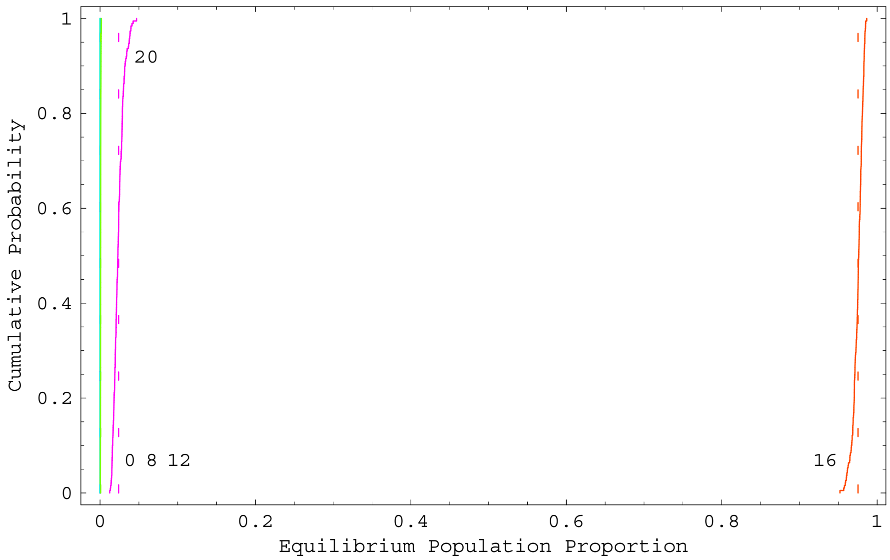

Figure 5.10: Sensitivity analysis for five-player $\mathrm{SAA}_{\lambda=2}$ game with strategies $20k \in \{0, 8, 12, 16, 20\}$. Compare to the replicator dynamics for the maximum likelihood payoff matrix for this game in Figure 5.5.

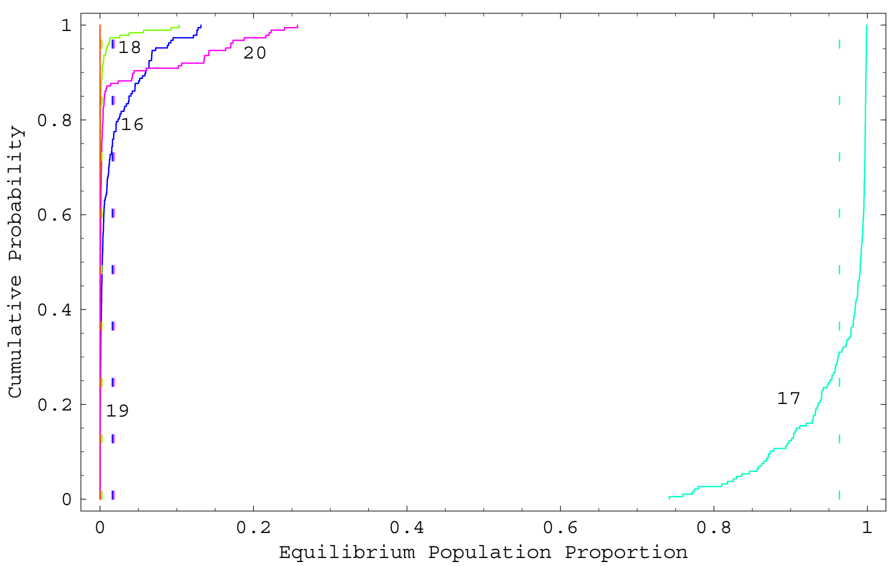

Figure 5.11: Sensitivity analysis for five-player $\mathrm{SAA}_{\lambda\sim E}$ game with strategies $20k \in \{16, \ldots, 20\}$. Compare to the replicator dynamics for the maximum likelihood payoff matrix for this game in Figure 5.6.

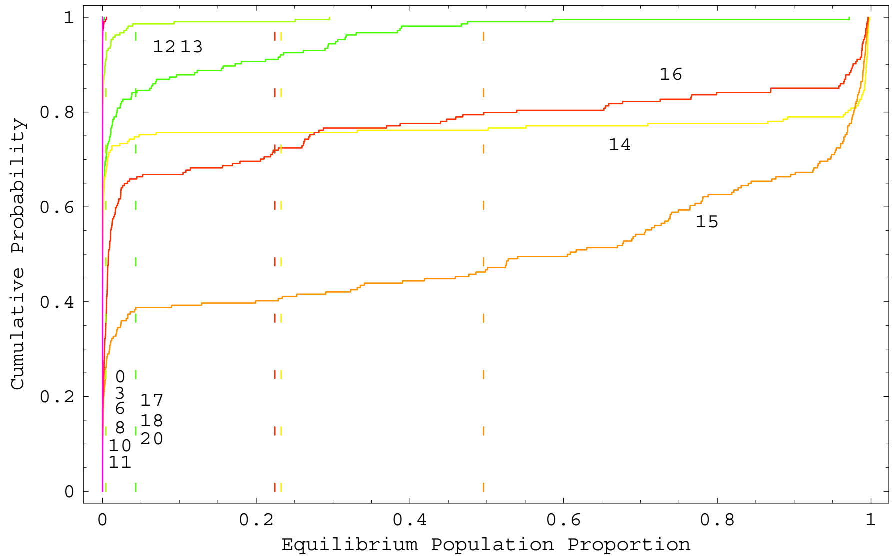

Figure 5.12: Sensitivity analysis for two-player $\text{SAA}_{\lambda\sim E}$ game with strategies $20k \in \{0,3,6,8,10,11,\ldots,17,18,20\}$. Strategy 15 is the only one with most of its mass significantly above zero. In Figure 5.8, 15 is the only strategy to survive.

more than two strategies). As before, each column records the average payoffs for each strategy in a particular profile. But for the two-strategy symmetric case, rather than display a payoff for each agent, we display the average payoffs for each strategy. The paired columns are given in lexicographic order from left to right, starting with all straightforward bidders: $\langle SB, SB, SB, SB, SB\rangle$. The second column is $\langle SB, SB, SB, SB, \text{PP}(\boldsymbol{\pi}^{\text{SB}})\rangle$, and so forth. In the first profile, each SB agent receives an expected payoff of about 1.4. In the second profile, each SB agent receives an expected payoff of about 1.5, while the sole $\text{PP}(\boldsymbol{\pi}^{\text{SB}})$ agent receives an expected payoff of about 2.3.

Determining Expected Prices in Equilibrium

Not to be confused with Walrasian equilibrium prices, we now consider the expected prices for an SAA game, given a particular symmetric mixed profile (typically a Nash equilibrium). We compute this by weighting the final prices for each realized (pure strategy) profile by the probability of obtaining that profile given the equilibrium mixed-strategy probabilities. Given an n-player game with S strategies and all agents playing the mixed strategy $\langle \alpha_1, \ldots, \alpha_S\rangle$, the probability of a particular profile $\langle n_1, \ldots, n_S\rangle$, where n_s is the number of players playing strategy s, is (paralleling Equation 3.10 for replicator dynamics where the mixture probabilities are replaced with population proportions)

$$\frac{n!}{n_1! \cdots n_S!} \cdot \alpha_1^{n_1} \cdots \alpha_S^{n_S}.$$

When there is a pure strategy equilibrium, we have the special case that all but one profile has zero weight.

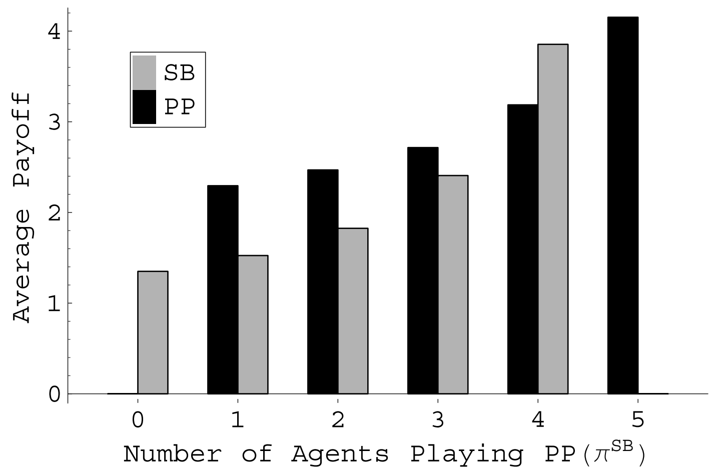

Figure 5.13: Payoff matrix for the game restricted to two strategies: SB and $\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{SB}})$. The gray bars show the payoffs to SB agents and the solid bars show payoffs to predicting agents. The profiles are denoted by the number of predictors in the profile, from zero to five. Average payoffs were estimated from 200 thousand simulated games for each of the six profiles.

We see from inspection of the SB vs. $\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{SB}})$ game in Figure 5.13 that $\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{SB}})$ is a dominant strategy since in every profile, the agents playing SB would do better switching to $\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{SB}})$. Thus it is also the unique equilibrium. The expected final prices are then estimated simply by the average final prices in this equilibrium profile: $\langle 11.2, 6.8, 3.8, 2.0, 0.8\rangle$. We summarize the results for this game (and those in subsequent sections) in Table 5.2.

Market Efficiency with Price Predictors

In addition to considering the perspective of agents bidding in SAA, we also consider how price prediction strategies affect market (or social) efficiency. That is, how is aggregate utility (including the sellers) affected by smarter strategies in SAA? The computation of efficiency for a given profile is analogous to that for determining expected prices above, using the expected efficiencies of each realized pure profile. The expected efficiency of a pure profile, then, is the expectation over the type distribution of aggregate utility when the profile is played. Finally, aggregate utility of an SAA outcome is the sum of the payoffs of the bidders plus the sum of the final prices (the payoff to the sellers).[3] The optimal allocation is the assignment of goods to agents that maximizes aggregate utility. (Note that this is unaffected by prices.) When all agents play $\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{SB}})$, market efficiency relative to the optimal allocation is 86%, which is 98% of the efficiency of the all-SB market. The average payoff in the equilibrium profile (again computed in general by appropriate mixing of the pure-strategy payoffs) is 4.2, which is 308% of the payoff to agents in the all-SB market. Thus, at a loss of only 2% in social efficiency, agents can improve their average performance by a factor of three if they use a simple price prediction based on average prices in an all-SB market.

[3]Equivalently, aggregate utility is the sum of the values that the agents get from their bundles, not counting what they pay.

Since all agents individually are better off in this equilibrium with price-predicting strategies, why is efficiency lower? The advantage to agents from using the $\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{SB}})$ price-predicting strategy is that they reduce the number of instances in which they are left paying for goods they cannot use (because they do not obtain their complete bundle of complementary goods). Although the exposure problem, as we showed above, can be very costly for *individual agents*, the allocation of these unused goods has no impact on social efficiency: if they are unused, it does not matter who gets the good. The payment by the agent is just a transfer to the resource seller, and our calculation of efficiency is indifferent between whether the buyer or the seller has the good or the money. Efficiency falls because to avoid the exposure problem, sometimes fewer bundles are completed, and so the allocation makes less valuable use of the available resources. In other words, price prediction prevents spurious purchases, improving buyers' payoffs at substantial cost to the seller, netting a slight loss in social efficiency.

5.5 How Does Price Prediction Help?

We have shown that baseline price prediction can substantially improve expected payoffs for bidding agents. We now explore the reason for the improvement: under what conditions does our price-predicting agent bid differently than SB, and how do these specific behavioral changes contribute to the improvement in expected payoffs?

Compared to SB, our price-predicting agents use the prediction vector to modify two behaviors: choosing the best bundle of goods on which to bid, and deciding whether to participate in the bidding at all. We could of course say both behaviors manifest a single decision: on which bundle to bid, with "don't participate" equivalent to bidding on the null bundle. We break the decision problem into participation and bundle selection because these two decisions are qualitatively different. We can then assess how these two aspects of the price prediction strategy improve performance.

Participation-Only Price Prediction

In order to decompose the effects reported in the previous section into those due to changes in participation and those due to changes in the choice of the bundle on which to bid, we construct a new bidding strategy. Agents first calculate the best bundle on which to bid using myopically perceived prices, as in SB (Section 4.4). Then they choose to participate in the current round of bidding only if surplus from that bundle, valued at the *predicted*, perceived prices (Equation 4.5), is positive. The predicted prices are used only for the participation decision, and not for selecting the best bundle on which to bid. We call this strategy $\mathrm{PP}_{\mathrm{po}}(\boldsymbol{\pi}^{x})$ where x labels an initial prediction vector.

In Figure 5.14 we present the payoff matrix for agents who choose either *participation-only* prediction, or straightforward bidding. The qualitative results for each possible strategy profile are similar to those in Figure 5.13. Again, the dominant (and therefore equilibrium) strategy is when all agents play $\mathrm{PP}_{\mathrm{po}}(\boldsymbol{\pi}^{\mathrm{SB}})$ with probability one. Expected payoff is 4.1, which is 2% lower than in the equilibrium with full-prediction agents as described above. Relative efficiency is slightly lower (half a percentage point).

To compare participation-only prediction to full prediction more thoroughly, we compute the ratio of payoff for participation-only prediction to payoff for full prediction for every possible environment of agents mixing between prediction and SB. We graph the results in Figure 5.15. The payoff to participation-only is less than to full-prediction as long as the probability of playing SB is less than 0.32. At best, if all agents predict, participation-only prediction achieves 98.1% of the

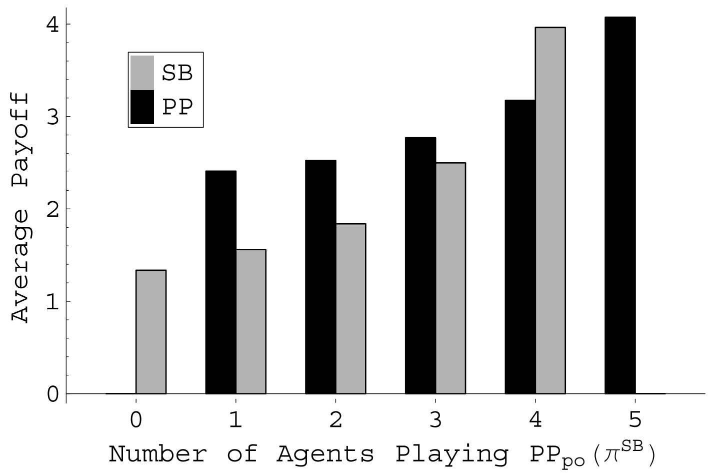

Figure 5.14: Payoff matrix when agents choose between SB and $\mathrm{PP}_{\mathrm{po}}(\boldsymbol{\pi}^{\mathrm{SB}})$.

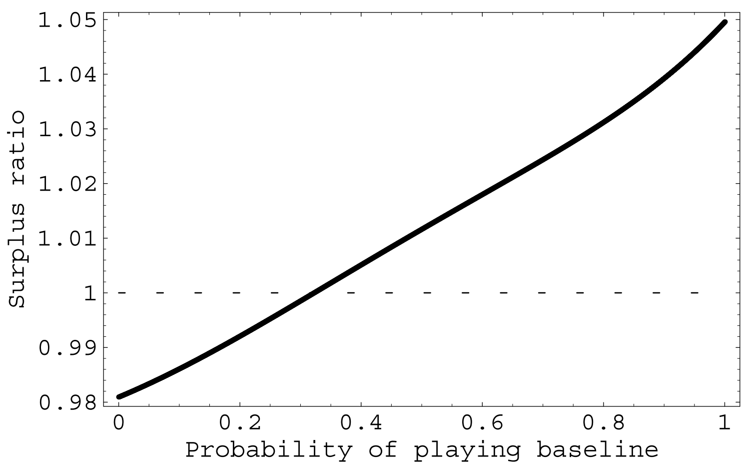

Figure 5.15: If agents play SB with probability less than 0.32, then full prediction does slightly better than participation-only prediction.

surplus of full prediction. At worst, in an environment of all SB agents, full prediction does slightly worse than participation-only prediction. Of course, we have found that in equilibrium agents will always use the prediction strategy.

Thus, we conclude that nearly all of the performance gain comes from the participation decision (that is, from sometimes dropping out earlier, which has the effect of reducing the risk of the exposure problem). In fact, far enough away from equilibrium the participation-only strategy secures more than 100% of the gains from price prediction.

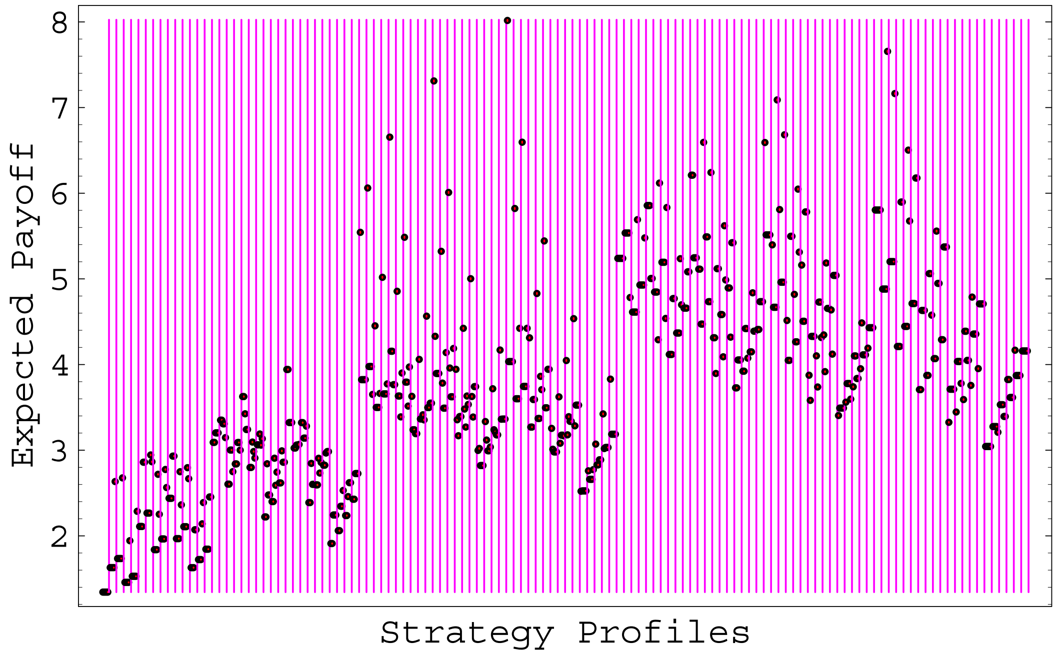

Figure 5.16: Payoff matrix for the game with four predicting strategies and SB. Each of the 126 columns (which are not meant to be distinguishable in this depiction) corresponds to a strategy profile: *all SB*, $\langle SB, SB, SB, SB, SB\rangle$, through *all* $\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{SB}})$ in lexicographic order, given the ordering $\langle SB$, $\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{EDPE}})$, $\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{EPE}})$, $\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{SC}})$, $\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{SB}})\rangle$. Payoffs were estimated by simulating 17 million games per profile.

5.6 Comparison of Point Predictors

Section 4.6 defines five point prediction strategies: SB ($= \mathrm{PP}(\mathbf{0})$), $\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{SB}})$, $\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{SC}})$, $\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{EPE}})$, and $\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{EDPE}})$. The five $\boldsymbol{\pi}^x$ initial prediction vectors corresponding to these strategies are given in Table 5.1. We consider the game in which each of the five agents can choose which of these five strategies to play. We present the resulting payoff matrix in Figure 5.16.

There are 126 possible combinations of five strategies among five players (each shown as a column in the payoff matrix). With this large problem, we did not find a dominant or pure strategy Nash equilibrium through inspection. Instead, we used replicator dynamics (Section 3.6) to solve for a symmetric mixed strategy equilibrium. In the equilibrium we found, agents play $\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{SC}})$ with probability 0.45, and $\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{EDPE}})$ with probability 0.55.

The average equilibrium prices are $\langle 10.6, 6.5, 4.0, 2.2, 0.91\rangle$. The expected payoff for an agent in the symmetric mixed strategy equilibrium is 4.3, which is 316% of the payoff in the all-SB market. The average efficiency of the allocation is 86%, which is 98% as efficient as the all-SB market. Just as in the $\mathrm{PP}(\boldsymbol{\pi}^{SB})$ vs. SB game, agents can triple their average performance at a cost of only a 2% loss in social efficiency by using price prediction (in this case, using a mixture of self-confirming and Walrasian predictions).

It may be surprising that all playing $\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{SC}})$—and so predicting perfectly—is not an equilibrium. A price predictor with a perfect prediction has an incentive to make its prediction worse. The

Prediction Methods	Predicted Final Price Vectors				
SB (= $\mathrm{PP}(\mathbf{0})$)	0	0	0	0	0
$\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{SB}})$	14.8	10.7	7.6	4.6	1.9
$\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{SC}})$	13.0	8.7	5.4	3.0	1.2
$\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{EPE}})$	26.0	14.2	6.9	2.5	0.3
$\mathrm{PP}(\boldsymbol{\pi}^{\mathrm{EDPE}})$	20.0	12.0	8.0	2.0	0.0

Table 5.1: Price predictions for five point prediction methods applied to $\mathrm{SAA}_{\lambda\sim U}$ with $n = m = 5$ and $V = 50$. Compare to realized average prices in Table 5.2.

explanation is that the predictor's performance is a function of how it uses the price prediction, as well as a function of its quality. In the next section we find that by using self-confirming *distribution* prediction instead of only point predictions for final prices, an agent can further improve its performance.

5.7 Self-Confirming Distribution Predictors

We now analyze the performance of self-confirming distribution predictors in a variety of SAA environments, against a variety of other strategies.

Environments and Strategy Space

The bulk of our computational effort went into the environment—$\mathrm{SAA}_{\lambda\sim U}$ ($n = m = 5$, $V = 50$)—analyzed in Sections 5.2 and 5.4. As described in Section 5.7, the empirical game for this setting provides much evidence supporting the unique strategic stability of $\mathrm{PP}(F^{\mathrm{SC}})$. We complement this most detailed trial with smaller empirical games for a range of other SAA environments. Altogether, we studied selected environments with uniform, exponential, and fixed distributions for desired bundle size; and agents in $3 \leq n \leq 8$; goods in $3 \leq m \leq 7$.

To varying degrees, we analyzed the interacting performance of 53 different strategies. These were drawn from the three strategy families described in Chapter 4: sunk-aware, point predictor, and distribution predictor. For each family we varied a defining parameter to generate the different specific strategies. Appendix D describes the details of these 53 strategies. We chose the strategies we deemed most promising but make no claim that we covered all reasonable variations. Our emphasis is on evaluating the performance of $\mathrm{PP}(F^{\mathrm{SC}})$ in combination with the other strategies. One of the noteworthy alternatives is $\mathrm{PP}(F^{\mathrm{SB}})$, which employs the price distribution prediction formed by estimating an empirical distribution of prices resulting from all agents playing SB.

In summary, we considered:

- 21 sunk-aware strategies, including SB ($k = 1$): $20k \in \{0, \dots, 20\}$.
- 13 point predictors, by varying the method used to generate the initial prediction vector. This includes a strategy that predicts all prices at ∞, meaning it never bids unless its desired bundle size (λ) is one.
- 19 distribution predictor strategies, by varying the method used to generate the distribution prediction. This includes degenerate and Gaussian distributions based on point prediction vectors (details in Appendix D).

Games (i.e., strategy sets)	Equilibrium Profiles	% Eff.	Payoff	Average Final Price Vectors				
$\{\text{SB}, \text{PP}(\boldsymbol{\pi}^{\text{SB}})\}$	all $\text{PP}(\boldsymbol{\pi}^{\text{SB}})$	86	4.15	11.2	6.8	3.8	2.0	0.77
$\{\text{SB}, \text{PP}_{\text{po}}(\boldsymbol{\pi}^{\text{SB}})\}$	all $\text{PP}_{\text{po}}(\boldsymbol{\pi}^{\text{SB}})$	85	4.07	11.8	6.9	3.7	1.7	0.58
$\{\text{SB}, \text{PP}(\boldsymbol{\pi}^{\text{SC}})\}$	all $\text{PP}(\boldsymbol{\pi}^{\text{SC}})$	88	3.05	13.0	8.7	5.4	3.0	1.17
$\{\text{SB}, \text{PP}(\boldsymbol{\pi}^{\text{SB}}), \text{PP}(\boldsymbol{\pi}^{\text{SC}}), \text{PP}(\boldsymbol{\pi}^{\text{EPE}}), \text{PP}(\boldsymbol{\pi}^{\text{EDPE}})\}$	0.45 SC, 0.55 EDPE	86	4.25	10.6	6.5	4.0	2.2	0.91
	Additional Profiles							
	all SB	87	1.35	14.8	10.7	7.6	4.6	1.90
	all $\text{PP}(\boldsymbol{\pi}^{\text{EPE}})$	74	5.80	4.7	2.1	1.7	1.2	0.55
	all $\text{PP}(\boldsymbol{\pi}^{\text{EDPE}})$	83	5.24	8.1	4.5	2.7	1.6	0.70

Table 5.2: Average final price vectors in $\text{SAA}_{\lambda\sim U}$ ($n = m = 5$ and $V = 50$), percent allocation efficiency relative to the global optimum, and average payoff for the (symmetric) equilibria of several restricted games as well as for other, non-equilibrium profiles. $\text{PP}_{\text{po}}(\boldsymbol{\pi}^{\text{SB}})$ refers to participation-only prediction (see Section 5.5) using the baseline prediction vector. Sensitivity analysis (Section 3.8) confirms that all equilibrium results are robust to sampling noise, and all reported efficiencies and payoffs have very small error bars.

With 53 strategies and five agents there are a daunting number of profiles: $\binom{n+S-1}{n} = \binom{57}{5}$ > 4 million. Using the brute-force method (Section 3.3) of estimating the expected payoffs for each profile by running millions of simulations of the auction protocol, estimating the entire payoff function is infeasible. However, it is straightforward to estimate the complete payoff matrix for various subsets of the 53 strategies. And as we describe below, we do not need the full payoff matrix to reach conclusions about equilibria in the 53-strategy game.

Analysis of $\mathrm{SAA}_{\lambda\sim U}$ with Five Agents and Five Goods

The largest empirical SAA game we have constructed is for the SAA environment with five agents, five goods, and the uniform valuations model. We have estimated payoffs for 4916 strategy profiles, out of the 4.2 million distinct combinations of 53 strategies. Payoff estimates are based on an average of 10 million samples per profile (though some profiles were simulated for as few as 200 thousand games, and some for as many as 200 million). Despite the sparseness of the estimated payoff function (covering only 0.1% of possible profiles), we obtained several results.

First, we conjectured that the self-confirming distribution predictor strategy, $\mathrm{PP}(F^{\mathrm{SC}})$, would perform well. We directly verified this: *the profile where all five agents play a pure* $\mathrm{PP}(F^{\mathrm{SC}})$ *strategy is a Nash equilibrium* in the game restricted to 53 strategies. No unilateral deviation to any of the other 52 pure strategies is profitable. To verify a pure-strategy symmetric equilibrium (all agents playing s) for n players and S strategies, one needs payoffs for only S profiles: one for each strategy playing with $n-1$ copies of s. The symmetric profile is an equilibrium if there are no profitable deviation profiles (i.e., obtained by changing the strategy of one player to obtain a higher payoff given the others).

The fact that $\mathrm{PP}(F^{\mathrm{SC}})$ is pure symmetric Nash for this game does not of course rule out the existence of other Nash equilibria. Indeed, without evaluating any particular profile, we cannot eliminate the possibility that it represents a (non-symmetric) pure-strategy equilibrium itself. However, the profiles we did estimate provide significant additional evidence, including the elimination of broad classes of potential symmetric mixed equilibria.

Let us define a strategy *clique* as a set of strategies for which we estimated payoffs for all combinations. Each clique defines a subgame, for which we have complete payoff information. Within our 4916 profiles we have eight maximal cliques, all of which include strategy $\mathrm{PP}(F^{\mathrm{SC}})$.[4] For each of these subgames, $\mathrm{PP}(F^{\mathrm{SC}})$ is the only strategy that survives iterated elimination of (strictly) dominated (pure) strategies.[5] It follows that $\mathrm{PP}(F^{\mathrm{SC}})$ is the unique (pure or mixed strategy) Nash equilibrium in each of these clique games. We can further conclude that in the full 53-strategy game there are no mixed strategy equilibria with support contained within any of the cliques, other than the special case of the pure-strategy $\mathrm{PP}(F^{\mathrm{SC}})$ equilibrium.

We can show that $\mathrm{PP}(F^{\mathrm{SC}})$ is the only small-support mixed strategy, among the two-strategy cliques for which we have calculated payoffs, that is even an approximate equilibrium. Of the $\binom{52}{2} = 1326$ pairs of strategies not including $\mathrm{PP}(F^{\mathrm{SC}})$, we have all profile combinations for 46. For these we obtained a lower bound of 0.32 on the value of ε (Section 3.1) such that a mixture of one of these pairs constitutes an ε-Nash equilibrium. In other words, for any symmetric profile defined by such a mixture, an agent can improve its payoff by a minimum of 0.32 through deviating

[4] One clique is a 10-strategy game with 2002 unique profiles; three are 5-strategy games (126 profiles each); one is a 4-strategy game (56 profiles); and three are 3-strategy games (21 profiles).

[5] Even better, $\mathrm{PP}(F^{\mathrm{SC}})$ is a dominant strategy in three of the subgames (two 3-strategy subgames and the one 5-strategy subgame in which it appears).

to some other pure strategy. For reference, the payoff for the all-$\mathrm{PP}(F^{\mathrm{SC}})$ profile is 4.51, so this represents a nontrivial difference.

Finally, for each of the 4916 evaluated profiles, we can derive a bound on the ε rendering the profile itself an ε-Nash pure-strategy equilibrium. The three most strategically stable profiles by this measure (i.e., lowest potential gain from deviation, ε) are:

1. all $\mathrm{PP}(F^{\mathrm{SC}})$: $\varepsilon = 0$ (confirmed Nash equilibrium)

2. one $\mathrm{PP}(F^{\mathrm{SB}})$, four $\mathrm{PP}(F^{\mathrm{SC}})$: $\varepsilon > 0.13$

3. two $\mathrm{PP}(F^{\mathrm{SB}})$, three $\mathrm{PP}(F^{\mathrm{SC}})$: $\varepsilon > 0.19$

The remaining profiles have $\varepsilon > 0.25$.

Our conclusion from these observations is that $\mathrm{PP}(F^{\mathrm{SC}})$ is a highly stable strategy within this strategic environment, and likely uniquely so. Of course, only limited inference can be drawn from even an extensive analysis of only one particular SAA environment.

Self-Confirming Prediction in Other Environments

To test whether the strong performance of $\mathrm{PP}(F^{\mathrm{SC}})$ generalizes across other SAA environments, we undertook smaller versions of this analysis on other models. Specifically, we explored 16 additional instances of SAA: eight each with the uniform ($\mathrm{SAA}_{\lambda\sim U}$) and exponential ($\mathrm{SAA}_{\lambda\sim E}$) models (3–8 agents, 3–7 goods).[6] For each we derived self-confirming price distributions (Section 4.6). We also derived price vectors and distributions for the other prediction-based strategies. For 11 of the environments (eight $\mathrm{SAA}_{\lambda\sim U}$ and three $\mathrm{SAA}_{\lambda\sim E}$), we evaluated 27 profiles: one with all $\mathrm{PP}(F^{\mathrm{SC}})$, and for each of 26 other strategies s, one with $n-1$ agents playing $\mathrm{PP}(F^{\mathrm{SC}})$ and one agent playing s. We ran between two and ten million games per profile in all of these environments.

For each of these 11 models, we identified the seven best responses to others playing $\mathrm{PP}(F^{\mathrm{SC}})$ (which invariably included $\mathrm{PP}(F^{\mathrm{SC}})$ itself). To economize on simulation time, for the other five $\mathrm{SAA}_{\lambda\sim E}$ environments we used the seven best responses found for the most similar of the simulated $\mathrm{SAA}_{\lambda\sim E}$ environments. We then evaluated all profiles involving these strategies (i.e., a 7-clique) in each of the 16 environments, based on at least 340,000 samples per profile.

We summarize our results in Table 5.3. We report the percentage gain for a participant that deviates from all-$\mathrm{PP}(F^{\mathrm{SC}})$ to the best of the 26 other strategies, i.e., $\varepsilon\%\left(\mathrm{PP}(F^{\mathrm{SC}})\right)$ (Definition 3.2). The next two columns report sensitivity information about this figure (see Section 3.8). We show the mean ε percentage, $\overline{\varepsilon\%\left(\mathrm{PP}(F^{\mathrm{SC}})\right)}$ based on 30 samples from the distribution representing our belief of the true expected payoff matrix. The close correspondence between $\varepsilon\%$ and $\overline{\varepsilon\%}$ indicates tight confidence bounds on our estimate of $\varepsilon\%$ using the maximum-likelihood payoff matrix. Additionally, we calculate the probability that all-$\mathrm{PP}(F^{\mathrm{SC}})$ is a Nash equilibrium, i.e., $\Pr\left(\varepsilon\left(\mathrm{PP}(F^{\mathrm{SC}})\right) = 0\right)$. Finally, for each environment with $n \le 6$ we use replicator dynamics (Section 3.6) to find a mixed strategy equilibrium for the 7-clique, and report the probability that an agent plays $\mathrm{PP}(F^{\mathrm{SC}})$ in this (generally mixed) Nash equilibrium.

In 15 out of these 16 environments, $\mathrm{PP}(F^{\mathrm{SC}})$ was verified to be an ε-Nash equilibrium for an ε less than 2% of the all-$\mathrm{PP}(F^{\mathrm{SC}})$ payoff. That is, no single agent could gain as much as 2% by deviating. The worst performance by $\mathrm{PP}(F^{\mathrm{SC}})$ is in environment $U(7,8)$, for which a strategy

[6] We additionally consider [Osepayshvili *et al.*, 2005] a model with fixed valuations (a game of complete information) specially constructed so that no self-confirming prediction exists. In this case, the $\mathrm{PP}(F^{\mathrm{SC}})$ strategy uses its best approximation to a self-confirming distribution. As expected, the strategy does poorly in this rather pathological environment.

$\mathrm{SAA}_{\lambda\sim *}(m,n)$	$\varepsilon\%\left(\mathrm{PP}(F^{\mathrm{SC}})\right)$	$\overline{\varepsilon\%}$	$\Pr(\varepsilon = 0)$	$\Pr\left(\mathrm{PP}(F^{\mathrm{SC}})\right)$
$E(3,3)$	0	0	1.00	1.00
$E(3,5)$	0	.09	.600	.996
$E(3,8)$	.83	.85	0	—
$E(5,3)$	0	0	1.00	.999
$E(5,5)$	0	.01	.900	.998
$E(5,8)$	.60	.64	0	—
$E(7,3)$	0	.06	.667	.992
$E(7,6)$	.04	.10	.567	.549
$U(3,3)$	1.24	1.26	0	.725
$U(3,5)$	0	0	1.00	1.00
$U(3,8)$	.56	.53	0	—
$U(5,3)$	1.35	1.35	0	.809
$U(5,8)$	1.59	1.62	0	—
$U(7,3)$	.81	.84	0	.942
$U(7,6)$	.52	.52	0	.929
$U(7,8)$	4.98	4.94	0	—

Table 5.3: Performance of $\mathrm{PP}(F^{\mathrm{SC}})$ as a candidate symmetric equilibrium for various $\mathrm{SAA}_{\lambda\sim U}$ and $\mathrm{SAA}_{\lambda\sim E}$ environments.

deviation could improve expected payoff by only 5%. Overall, we regard this as favorable evidence for the $\mathrm{PP}(F^{\mathrm{SC}})$ strategy across a range of SAA environments.

5.8 Conclusion

In our experiments in sunk-awareness, we found for the restricted game in which agents can choose only the sunk-awareness parameter, k, one regularity across SAA environments: the greater the number of agents, the less sunk aware an agent should be. Otherwise, the equilibrium k depended sensitively on the SAA environment, in particular on the agent type distribution.

We then turned to the analysis of price prediction strategies and found that even a simple point price prediction can greatly improve performance (at a small cost in market efficiency) by avoiding the exposure problem. In fact, we found that (at least for the environment we investigated) the binary choice of participation or not had a greater contribution to improvement than the use of predictions to guide choice of which goods to pursue. Still, point price prediction is a suboptimal strategy and so we considered next distribution predictors.

Our final class of trading strategies for SAA environments with complementarities place bids based on probabilistic predictions of final prices. Like the approach of Greenwald and Boyan [2004], our policy tackles the exposure problem head-on, by explicitly weighing the risks and benefits of placing bids on alternative bundles, or no bundle at all. The specific strategy generalizes the point prediction method, and like that scheme is parameterized by the *method* for generating predictions. By explicitly conditioning on context (e.g., type distributions), such trading strategies are potentially robust across varieties of SAA environments.

The strategy we consider most promising employs what we call *self-confirming price distributions*. A price distribution is self-confirming if it reflects the prices generated when all agents play the trading strategy based on this distribution. Although such self-confirming distributions may not always exist, we expect they will (at least approximately) in many environments of interest, especially those characterized by relatively diffuse uncertainty and a moderate number of agents. An iterative simulation algorithm, which we derive elsewhere [Osepayshvili *et al.*, 2005], appears effective for deriving such distributions.

To assess the self-confirming distribution predictor, we explore a range of sunk-aware and price prediction strategies described in Chapter 4. Despite the infeasibility of exhaustively exploring the profile space, our analyses support several game-theoretic conclusions. The results provide favorable evidence for our favored strategy—very strong evidence in one environment we investigated intensely, and somewhat less categorical evidence for a range of variant environments.

Neither the self-confirming distribution predictor nor any other strategy is likely to be universally best across SAA environments. Nevertheless, our results establish the self-confirming price prediction strategy as the leading contender for dealing broadly with the exposure problem. If agents make optimal decisions with respect to price distributions that turn out to reflect the true underlying uncertainty given the strategies and the environment, there may not be room for performing a lot better. On the other hand, there are certainly areas where improvement should be possible, for example:

- accounting for one's own effect on prices, as in strategic demand reduction [Weber, 1997],
- incorporating price dependencies with reasonable computational effort, and
- timing of bids: trading off the risk of premature quiescence with the cost of pushing prices up.

Exploring these opportunities is the subject of future work.

Chapter 6

Taming TAC: Searching for Walverine

IN WHICH *we employ our empirical game methodology to choose the strategy for our agent in the Trading Agent Competition travel-shopping game.*

In Chapter 4 I describe key elements of our entry in the annual Trading Agent Competition (TAC). Focusing on the SAA-like subproblem of TAC hotel bidding, I describe price prediction as well as strategic bid shading for hotels. The previous chapter applies our empirical game methodology of Chapter 3 to the SAA game. Here we apply the methodology to TAC. A single TAC game requires close to eleven minutes of real time to simulate. Multiple times that amount of cpu time is needed when all eight agents are playing parametric variations of our agent, Walverine. This is in contrast to the milliseconds or less of real and cpu time for SAA. Thus, for TAC we have had to dig deeper into our collection of empirical game techniques. We begin with an analysis of bid shading in which we are able to estimate the full payoff matrix for a game restricted to two strategies: applying Walverine's bid shading strategy vs. bidding marginal values (no shading). We then consider a richer parameterization of Walverine's strategy, from which we choose 40 variations of Walverine. Applying control variates (Section 3.4) to the TAC game to reduce the variance of the sampled payoffs we additionally apply the technique of player reduction (Section 3.5) and find that the 4-player and 2-player reductions of the TAC game are far more manageable and likely entail little loss of fidelity. I report on our use of these techniques in choosing our strategy for Walverine in TAC Travel 2005.

6.1 Preliminary Strategic Analysis: Shading vs. Non-Shading

In this section, we investigate the strategic behavior of hotel bid shading (Section 4.10) in isolation from other strategic elements. To evaluate this effect, we define a variant non-shading strategy implemented by Walverine 2002 with its optimal bidding procedure (i.e., shading) turned off. Agents in this version bid their true (estimated) marginal values for every hotel room, based on predicted prices (Section 4.10). Our hypothesis was that this would hurt agent performance but at the same time improve social welfare. We in fact found, in a trial of 55 games with all non-shading Walverines, the market achieved slightly better efficiency than all shading Walverines.[1] The actual effect of varying strategy, however, will in general depend on the strategies of other agents.

[1] Though bid shading benefits the individual agent (by design) it degrades market efficiency because the market does not generally have available faithful signals of the relative value of goods to the various agents. If all agents shaded proportion-

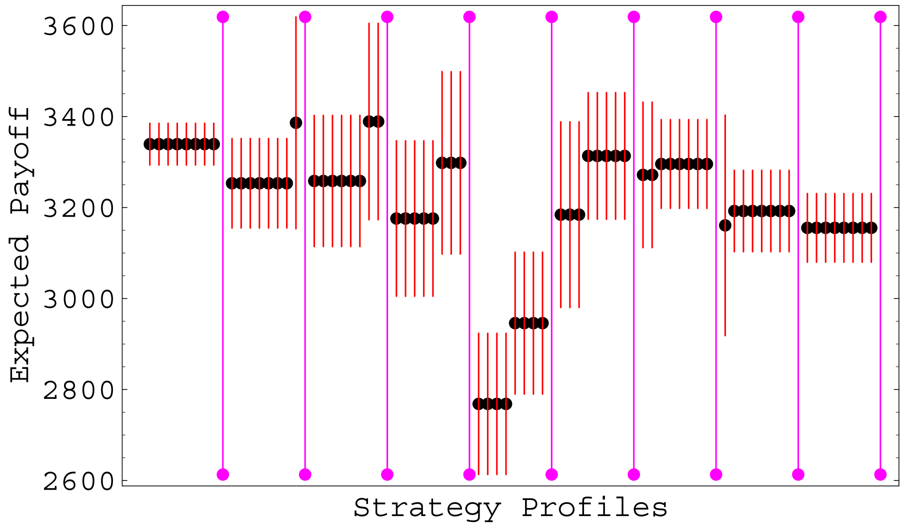

Figure 6.1: Empirical payoff matrix for shading vs. non-shading Walverine, based on roughly 100 games per profile. The profile of all shaders is on the left and all non-shaders on the right.

Figure 6.1 shows the empirical payoff matrix for the 8-player, 2-strategy game in which the available strategies are to play Walverine's strategy with shading on or off. Applying replicator dynamics (Section 3.6), we identify a symmetric mixed strategy equilibrium where agents shade with probability 0.11, and refrain from shading with probability 0.89. GAMBIT (Section 2.1) was unable to find a symmetric equilibrium after some hours of cpu time, finding only the asymmetric equilibrium in which five agents shade and three do not.

The predominance of truthful bidding in equilibrium demonstrates that this policy has advantages in a population with substantial shaders. It is not dominant, however, which means that as the population approaches all non-shaders, there is benefit to shading. In equilibrium, the payoffs to shading and non-shading agents are the same, each a best response to the given mixture. The average payoffs[2] for all shading, all non-shading, and the equilibrium mixed strategy are 3339, 3155, and 3209, respectively. (The corresponding market efficiencies are 88.5%, 89.4%, and 89.2%.) Playing the equilibrium strategy results in an average payoff gain of 53 per agent per game (but a loss of 46 in social welfare compared to all non-shaders).

The above results of course apply specifically to the version of shading employed in Walverine; alternative shading policies incorporated in other agent strategies may produce varying outcomes. Nonetheless, having no basis to speculate on the strategic interaction of shading in an environment with non-Walverine agents, we in fact used this analysis to choose our strategy in TAC-03. In particular, we played the symmetric mixed strategy equilibrium of the empirical payoff matrix of the restricted game: shading with 0.11 probability.

ally, then the relative offer prices would still provide the relevant information to the market. In general, however, the price reductions do not cancel out in this way, as the bidder's optimization includes many agent-specific contextual factors. We analyze these and other effects on TAC market efficiency elsewhere [Wellman *et al.*, 2003a].

[2]These were adjusted using a preliminary, ad hoc application of control variates, not described here, but subsumed by the approach in Section 6.3.

Next, we consider a richer parameterization, applied to Walverine's strategy in 2004, and the far more extensive experimentation required to explore this parameter space.

6.2 Walverine Parameters

The full TAC strategy space includes all policies for bidding on flights, hotels, and entertainment over time, as a function of prior observations (including own client preferences and ticket endowments—i.e., own type). To focus our search, we restrict attention to variations on our basic Walverine strategy [Cheng *et al.*, 2005], as originally developed for TAC-02 and refined incrementally for 2003, 2004, and 2005. Choosing the strategy for 2005 (after a preliminary attempt for 2004) has consisted primarily of applying our empirical game-theoretic approach to searching Walverine's parameter space (Section 6.5).

We illustrate some of the possible strategy variations by describing some of the exposed parameters. To invoke an instance of Walverine, the user supplies parameter values specifying which versions of the agent's modules to run, and what arguments to provide to these modules.

Flight Purchase Timing

Flight prices follow a biased random walk, as described in Section 4.7. Whereas flight prices are designed to increase in expectation given no information about the hidden parameter, conditional on this parameter prices may be expected to increase, decrease, or stay constant. Walverine maintains a distribution $\Pr(x)$ for each flight, initialized to be uniform on [-10,30], and updated using Bayes's rule given the observed perturbations Δ at each iteration: $\Pr(x \mid \Delta) = \alpha \Pr(x) \Pr(\Delta \mid x)$, where α is a normalization constant. Given this distribution over the hidden x parameter, the expected perturbation $E[\Delta' \mid x]$ for the next iteration is simply $(lb + ub)/2$, with bounds given by Equation 4.7. Averaging over the distribution for x, we have $E[\Delta'] = \sum_x \Pr(x) E[\Delta' \mid x]$.

Given a set of flights that Walverine has calculated to be in its optimal bundle of travel goods, it decides which to purchase now as a function of the expected perturbations, current holdings, and marginal flight values. On a high level, the strategy is designed to defer purchase of flights that are not quickly increasing, allowing for flexibility in avoiding expensive hotels as hotel price information is revealed. The flight purchase strategy can be described in the form of a decision tree as depicted in Figure 6.2. First, Walverine compares the expected perturbation ($E[\Delta']$) with a threshold $T1$, deferring purchase if the prices are not expected to increase by $T1$ or more. If $T1$ is exceeded, Walverine next compares the expected perturbation with a second higher threshold, $T2$, and if the prices are expected to increase by more than $T2$ Walverine purchases all units for that flight that are in its best bundle (given predicted prices).

If $T1 < E[\Delta'] < T2$, the Walverine flight delay strategy is designed to take into account the potential benefit of avoiding travel on high demand days. Walverine checks whether the flight constitutes one end of a *reducible* trip: one that spans more than a single day. If the trip is not reducible, Walverine buys all the flights. If reducible, Walverine considers its own demand (defined by the best bundle) for the day that would be avoided through shortening the trip, equivalent to the day of an inflight, and the day before an outflight. If our own demand for that day is $T3$ or fewer, Walverine purchases all the flights. Otherwise (reducible and demand greater than $T3$), Walverine delays the purchases, except possibly for one unit of the flight instance, which it will purchase if its marginal surplus exceeds another threshold, $T4$.

Though the strategy described above is based on sound calculations and tradeoff principles, it is difficult to justify particular settings of threshold parameters without making numerous assumptions

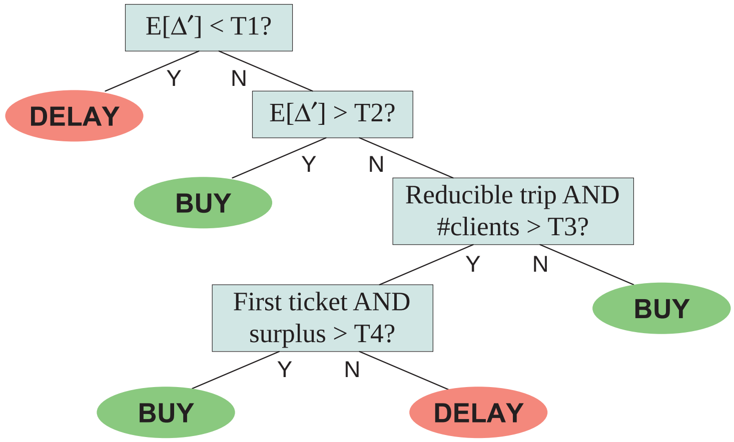

Figure 6.2: Decision tree for deciding whether to delay flight purchases.

and simplifications. Therefore we treat these as strategy parameters, to be explored empirically, along with the other Walverine parameters.

Bid Shading

Walverine's "optimal" bidding strategy (Section 4.10) identifies, for each hotel auction, the bid value maximizing expected utility based on a model of other agents' marginal value distributions. Because this optimization is based on numerous simplifications and approximations, we have added several parameters to control its use.

We first introduce a *shading mode* parameter, with possible values $\{0, 1, 2, 4\}$. As in Section 6.1, one choice is to turn off shading altogether and bid marginal values. This corresponds to shading mode 0. When shading mode is set to 4, Walverine bids a fixed fraction below marginal value. Another parameter, *shade percentage*, which only has effect if shade mode is 4, specifies this fixed fraction. There are two modes corresponding to the optimal shading algorithm, differing in how they model the other agents' value distributions. In the first (shade mode 1), the distributions are derived from the simple competitive model described in Section 4.10. For this mode, another parameter, *shade model threshold* turns off shading in case the model appears too unlikely given the price quote. Specifically, we calculate the probability that the 16th highest bid is greater than or equal to the quote according to the modeled value distributions, and if too low we refrain from using the model for shading. For the second optimal shading method (shade mode 2), instead of the competitive model we employ empirically derived distributions keyed on the hotel closing order.

Other Parameters

Walverine predicts hotel prices based on Walrasian price equilibrium analysis (Section 4.9) and forms crude price distributions using a hedging model (Section 4.9) with *outlier probability* π. Originally, Walverine used a setting of 0.06 for this parameter. Setting it to zero turns off hedging altogether (equivalently, a degenerate distribution prediction yielding point predictions with certainty).

Given a price distribution, one could optimize bids with respect to the distribution itself, or with respect to the *expected* prices induced by the distribution. (The two are of course equivalent when $\pi = 0$.) Although the former approach is more accurate in principle, necessary compromises in implementation render it ambiguous in practice which produces superior results [Greenwald and Boyan, 2004; Stone *et al.*, 2003; Wellman *et al.*, 2004]. Thus, we include a parameter controlling which method to apply in Walverine.

Several agent designers have reported employing *priceline* predictions, accounting for the impact of one's own demand quantity on price. We implemented a version of the *completion algorithm* [Boyan and Greenwald, 2001] that optimizes with respect to pricelines, and included it as a Walverine option. A further parameter selects how price predictions and optimizations account for outstanding hotel bids in determining current holdings. In one setting current bids for open hotel auctions are ignored, and in another the current hypothetical winnings are treated as actual holdings. This is the *expected holdings* parameter.

We choose among a discrete set of policies for trading entertainment. As a baseline, we implemented the strategy employed by LivingAgents in TAC-01 [Fritschi and Dorer, 2002]. We also applied reinforcement learning to derive policies from scratch, expressed as functions of marginal valuations and various additional state variables. The policy employed by Walverine in TAC-02 was derived by Q-learning over a discretized state space. For TAC-03 we learned an alternative policy, this time employing a neural network to represent the value function. Finally, we have also implemented an entertainment trading policy based on the top-scoring Whitebear agent [Vetsikas and Selman, 2003].

Agent Module Version Parameters

Beginning in 2004, a set of TAC rule changes reduced the viability of strategies dictating the early purchase of all flights. These changes included shortening the time delay to the first hotel closing from from four minutes to one, increasing the frequency of flight price perturbations to every 10 seconds, and lowering the range from which these expected perturbations are drawn (we describe the current flight rules in Section 4.7 while our full description of Walverine includes the original rules [Cheng *et al.*, 2005]). These changes simplified the problem of inferring the value of x in predicting future flight prices, while reducing the ex ante expected flight price perturbations. The expected perturbation of a flight price at time t could now even be negative given observed price perturbations leading up to time t. Before these rule changes, gaining hotel price information was the only incentive to delaying flights. The rule changes for 2004 drastically reduced the cost of gaining this information and introduced the possibility of realizing lower flight costs at later times in the game.

To take advantage of these rule changes, we modified the flight module to maintain a distribution over the hidden parameter x based on observed price perturbations. The flight purchase algorithm described above was added, generating flight purchase decisions based on the expected value of these perturbations. The flight module version is used to select among different implementations of this algorithm, with newer revisions correcting errors found in early implementations. Flight version 0 will purchase all flights in the beginning of the game, as Walverine originally did [Cheng *et al.*, 2005]. The first implementation of the flight delay algorithm was found to have errors sufficient to render it useless and has been removed from use (flight version 1). Versions 2 and 3 implement the same algorithm, with the latter implementation taking flight price boundary conditions into account when computing expected perturbations of flight prices.

Choices for the hotel bidding module are currently restricted to the the five most recent versions (39–43) where newer versions both implement bug fixes and introduce new parameters and associated behavior options. We attempt to introduce parameters such that new versions are able to implement a strategy set that is a superset of the strategy set of older versions, where the use of multiple versions is a reflection of our failure to fully meet this goal. Version 40 increased the flight price refresh rate from every minute to every 10 seconds, and changed the way in which the $T4$ parameter is used. Version 42 corrected the way in which expected flight price perturbations are calculated by taking boundary conditions into consideration. Version 41 corrected a bug introduced in version 40, and similarly version 43 corrected a bug introduced in version 42. Due to these bugs, versions 40 and 42 exhibit degraded performance with respect to the updated versions.

The entertainment trading strategies are divided into four main classes, two of which feature strategies learned through game play and two of which are based on written descriptions of strategies from other agents. Of the learned policies, version 10 employs neural networks, while strategy 11 uses q-tables, both of which were trained using a combination of self-play and tournament play [Cheng *et al.*, 2005]. Versions 14 and 15 implement corrections to versions 11 and 12, respectively, where the former versions failed to transmit entertainment holdings to the optimization module used by the hotel agent. Version 12 is based on the description of entertainment trading featured in TAC competitor LivingAgents, with version 13 implementing the same correction necessary for learned policies 10 and 11. Versions 16 and 17 are both based on the strategy employed by whitebear, where version 16 had severe implementation errors and was removed from use soon after being incorporated into the strategy space.

Despite the presence of multiple unambiguously inferior strategies, we do not purge these from our database of games (i.e., our partial payoff matrix) because strategy profiles with some poor strategies can still prove useful in our game theoretic analysis, described below. And we have even occasionally been surprised to find theoretically inferior strategy performing well in our simulations.

6.3 Control Variates for the TAC Game

From the first years of the competition, we have used various client-preference adjustments for scores in TAC Travel to adjust for luck in the game, reducing variance and getting more statistically meaningful comparisons from fewer games. Here we present a formula derived for the latest TAC Travel rules, following the method of control variates, described in Section 3.4. For TAC, the function (game simulator) to which we apply control variates produces scores (payoffs) as a function of client preferences, other random game data, and agent strategies. Random factors in the game (Nature's and agents' types) include hotel closing order, flight prices, entertainment ticket endowment, and, most critically, client preferences. To apply the method we need to find summary measures of the random elements of the game that correlate with score. For example, an agent whose clients had anomalously low hotel premiums would have its score adjusted upward as a handicap. Or in a game with very cheap flight prices, all the scores would be adjusted downward to compensate. The specific control variables we employ are the following, for a hypothetical agent A:

- x_1: Sum of A's clients' entertainment premiums ($8 \cdot 3 = 24$ values). $E[x_1] = (0+200) \cdot 3 \cdot 8 = 2400$.

- x_2: Sum of initial flight quotes (8 values; same for all agents). $E[x_2] = (250 + 400)/2 \cdot 8 = 2600$.

- x_3: Weighted total demand: Total demand vector (for each night, the number of the 64 clients who would be there that night if they got their preferred trips) dotted with the demand vector for A's clients. $E[x_3] = 540.16$ (see below).
- x_4: Sum of A's clients' hotel premiums (8 values). $E[x_4] = (50 + 150)/2 \cdot 8 = 800$.

The expectations are determined analytically. To do this for x_3 we note that a set of n clients' aggregate demand for a particular day is a random variable Z_n^p with a binomial distribution where the probability parameter is $p = p_1 = 4/10$ for days 1 and 4 and $p = p_2 = 6/10$ for days 2 and 3.[3] The n parameter is the number of clients in the set (8 for one agent and $(8-1) \times 8 = 56$ for the remaining seven agents). $E[x_3]$ is then

$$\begin{aligned} & 2E[(Z_8^{p_1})^2 + Z_8^{p_1} * Z_{56}^{p_1}] + 2E[(Z_8^{p_2})^2 + Z_8^{p_2} * Z_{56}^{p_2}] \\ =& 2\left(8 \cdot 56 p_1^2 + \sum_{i=0}^{8} i^2 \binom{8}{i} p_1^i (1-p_1)^{8-i}\right) + 2\left(8 \cdot 56 p_2^2 + \sum_{i=0}^{8} i^2 \binom{8}{i} p_2^i (1-p_2)^{8-i}\right) \\ =& 16\left(63 p_1^2 + p_1 + 63 p_2^2 + p_2\right) \\ =& 540.16. \end{aligned}$$

(We confirmed this derivation via Monte Carlo sampling, which we note is a perfectly adequate method for estimating the expectation of a control variate in cases where we cannot derive it analytically.)

Given the above, we adjust an agent's score by subtracting

$$\boldsymbol{\beta} \cdot (\boldsymbol{x} - E[\boldsymbol{x}]) = \sum_{i=1}^{4} \beta_i (x_i - E[x_i])$$

where the βs are determined by performing a multiple regression from $\boldsymbol{x}$ to score using a data set consisting of 2190 all-Walverine games. Using adjusted scores in lieu of raw scores reduces overall variance by 22% based on a sample of 9000 all-Walverine games.

We have also estimated the coefficients based on the 107 games in the TAC Travel 2004 semi-finals and finals and have proposed these as the basis for official score adjustments for the competition:

- $\beta_1 = 0.349$
- $\beta_2 = -1.721$
- $\beta_3 = -2.305$
- $\beta_4 = 0.916$

Note that we can see from these coefficients that it improves an agent's score somewhat to have clients with high entertainment premiums, it hurts performance to be in a game with high flight prices, it hurts to have clients that prefer long trips (particularly when other agents' clients do as well), and finally, having clients with high hotel premiums improves score.

Applying the score adjustment formula to the 2004 finals yields a reduction in variance of 9%. Table 6.1 shows the average raw scores and the average adjusted scores. Figure 6.3 shows the

[3] This follows from the client preference distribution, described in detail at `http://www.sics.se/tac`.

adjusted scores with error bars. Finally, Table 6.2 shows the p-values for the mean difference tests for all pairwise agent comparisons. Note that there is little doubt as to the best agent (whitebear [Vetsikas and Selman, 2003]).

Agent	Raw Score	Adjusted Score	95% C.I.
whitebear04	4122	4210	± 206
Walverine	3849	3921	± 150
LearnAgents	3737	3838	± 187
SICS02	3708	3816	± 174
NNN	3666	3698	± 145
UMTac-04	3281	3371	± 152
Agent@CSE	3263	3347	± 349
RoxyBot	2015	2117	± 468

Table 6.1: Scores, adjusted scores, and 95% mean confidence intervals on adjusted scores for the 35 games of the TAC Travel 2004 finals.

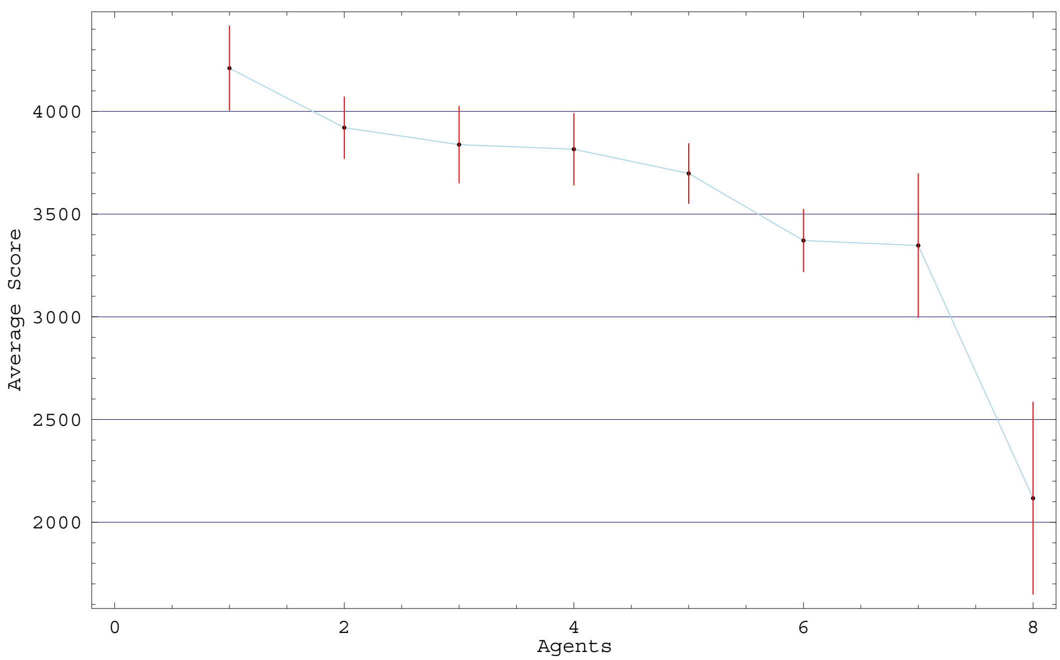

Figure 6.3: Comparison of the eight agents in the TAC-04 finals.

6.4 Player-Reduced TAC Experiments

As the first step in applying our empirical game methodology to TAC, we manually selected 40 distinct strategies, covering variant policies for bidding on flights, hotels, and entertainment (Sec-

	Walv.	LearnA.	SICS	NNN	UMTac	Agent@CSE	RoxyBot
whitebear04	0.012	0.004	0.002	0	0	0	0
Walverine	-	0.244	0.178	0.017	0	0.002	0
LearnAgents	-	-	0.429	0.116	0	0.007	0
SICS02	-	-	-	0.147	0	0.009	0
NNN	-	-	-	-	0.001	0.033	0
UMTac-04	-	-	-	-	-	0.449	0
Agent@CSE	-	-	-	-	-	-	0

Table 6.2: Mean difference tests for adjusted agent scores in TAC-04 finals.

tion 6.2). The full list of strategies is presented in Table 6.3. We have collected data for a large number of games: over 47,000 and counting, representing over thirteen of (almost continuous) simulation.[4] Each game instance provides a (control variate adjusted) sample payoff vector for a profile over our restricted strategy set.

Table 6.4 shows how our dataset is apportioned among the 1-, 2-, and 4-player reduced games. We are able to exhaustively cover the 1-player game, of course. We could also have exhausted the 2-player profiles, but chose to skip some of the less promising ones (29%) in favor of devoting more samples elsewhere. The available number of samples could not cover the 4-player games, but as we see below, even 1–2% is sufficient to draw conclusions about the possible equilibria of the game. Spread over the 8-player game, however, 47,000 instances would be insufficient to explore much, and so we refrain from any sampling of the unreduced game (except in the 2-strategy case in Section 6.1.)

In the spirit of hierarchical exploration, we sample more instances per profile as the game is further reduced, obtaining more reliable statistical estimates of the coarse background relative to its refinement. On introducing a new profile we generate a minimum required number of samples, and subsequently devote further samples to particular profiles based on their potential for influencing our game-theoretic analysis. The sampling policy employed was semi-manual and somewhat *ad hoc*, driven in an informal way by analyses of the sort described below on intermediate versions of the dataset. Developing a fully automated and principled sampling policy is the subject of future research.

We next present the results of our analysis in the 1-, 2-, and 4-player reduced TAC games. For the statistical comparisons here, it is important to point out the danger of allowing the results of statistical significance tests to influence the number of samples, as we have done to some extant. For example, given two identical distributions, if samples are collected until a mean difference test declares them different at some significance level, the test will eventually spuriously succeed. On the other hand, it is intuitively clear that when the underlying distributions are different it is very much possible to sample until the difference has been established beyond doubt.

[4]Our simulation testbed comprises two dedicated workstations to run the agents, another RAM-laden four-CPU machine to run the agents' optimization processes, a share of a fourth machine to run the TAC game server, and background processes on other machines to control the experiment generation and data gathering.

	Hotel Ver.	Shade Mode	Shade %	Flight Ver.	T1	T2	T3	T4	Ent. Ver.
1	39	1	-	0	-	-	-	-	11
2	39	1	-	0	-	-	-	-	12
3	39	1	-	1	25	50	200	3	11
4	39	1	-	1	25	100	200	3	11
5	39	1	-	1	50	50	0	0	11
6	39	1	-	1	50	100	100	3	11
7	39	1	-	1	50	100	200	3	11
8	39	1	-	1	50	100	200	3	12
9	39	1	-	1	50	200	200	3	11
10	39	1	-	1	50	200	200	3	12
11	40	1	-	2	25	100	200	3	11
12	40	1	-	2	25	200	200	3	11
13	40	1	-	2	50	50	0	0	11
14	40	1	-	2	50	100	200	3	11
15	40	1	-	2	50	200	200	3	11
16	41	1	-	2	25	50	200	3	11
17	41	1	-	2	25	100	200	3	11
18	41	1	-	2	50	50	0	0	11
19	41	1	-	2	50	50	0	0	15
20	41	1	-	2	50	100	100	2	14
21	41	1	-	2	50	100	100	3	11
22	41	1	-	2	50	100	100	3	15
23	41	1	-	2	50	100	200	2	11
24	41	1	-	2	50	100	200	3	11
25	41	1	-	2	50	100	200	3	13
26	41	0	-	2	50	100	200	3	15
27	41	1	-	2	50	100	200	3	15
28	41	1	-	2	100	100	0	0	11
29	42	1	-	3	25	100	200	3	11
30	42	1	-	3	50	125	200	3	11
31	42	1	-	3	50	150	200	1	11
32	42	4	20	3	50	150	200	3	15
33	42	0	-	3	75	150	200	2	11
34	42	4	50	3	75	150	200	2	11
35	43	1	-	3	50	125	200	3	11
36	43	1	-	3	50	125	200	3	16
37	43	1	-	3	50	125	200	3	17
38	43	0	-	3	75	150	200	2	11
39	43	1	-	3	75	150	200	2	17
40	43	4	10	3	75	150	200	3	17

Table 6.3: Candidate strategies (variants of Walverine) for TAC game analysis. All 40 strategies have $\pi = 0$, use non-priceline predictions, use expected holdings, and have a shade model threshold of 2%.

p	Profiles			Samples/Profile	
	total	evaluated	%	min	mean
8	> 314M	2114	0	12	22.3
4	123,410	2114	1.7	12	22.3
2	820	586	71.5	15	31.7
1	40	40	100	25	86.5

Table 6.4: Profiles evaluated in reduced TAC games (TAC$\downarrow_p$).

1-Player Game

The 1-player game (TAC$\downarrow_1$) would typically not merit the term "game", as it assumes each strategy plays only among copies of itself. Thus, its analysis considers no strategic interactions. To "solve" the game, we simply evaluate which strategy has the greatest expected payoff. For this experiment, we obtained 25–275 samples of each of the 40 1-player profiles, one for each strategy.

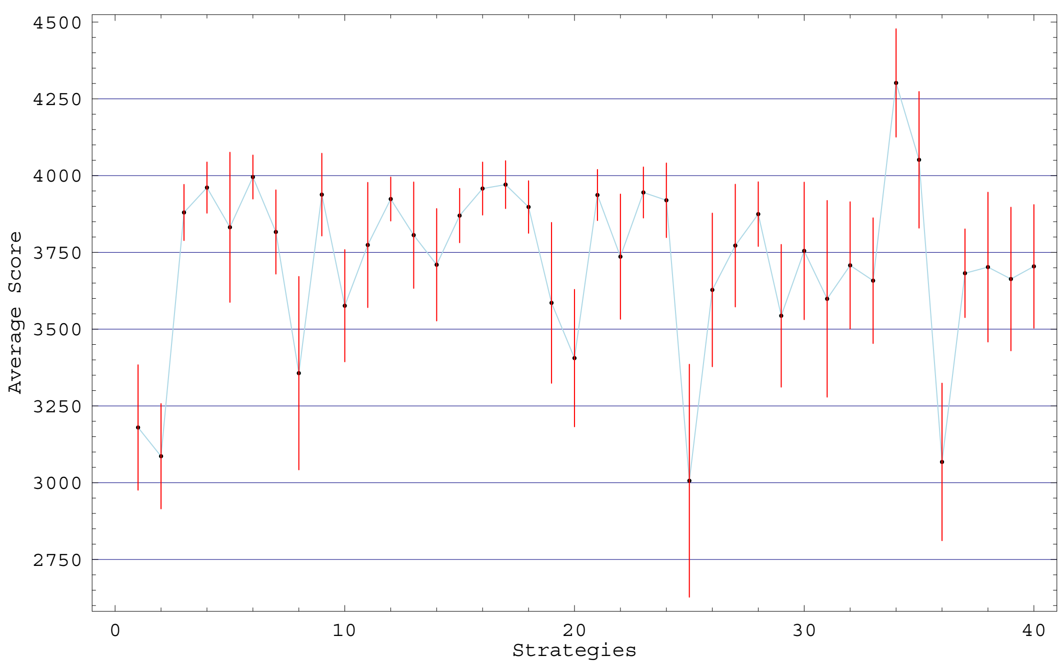

Figure 6.4: Average payoffs for (symmetric) strategy profiles in TAC$\downarrow_1$. Error bars delimit 95% confidence intervals.

Figure 6.4 displays the average payoffs for each 1-player profile. We tended to take more samples of the more promising profiles, but cannot statistically distinguish every profile in the ranking. Nevertheless our top strategy, number 34, performs dramatically—250 points—better than the rest. Pairwise mean-difference tests rank 34 above all others at significance levels of $p < 0.05$, and at $p < 0.01$ for all strategies but the next best, 35.

In the absence of further data, we might propose strategy 34, the unique pure-strategy Nash equilibrium (PSNE) of the 1-player game. In fact, however, this strategy was designed expressly to do well against itself: it shades all hotel bids by a fixed 50% rate, resulting in very low hotel prices. The profile is quite unstable, as an agent who shades less can get much better hotel rooms, but still benefit from the low prices. By exploring a less extreme reduction we can start to consider some of the strategic interactions.

2-Player Game

The two-player game, TAC$\downarrow_2$, comprises 820 distinct profiles: $\binom{40}{2} = 780$ where two different strategies are played by four agents each, plus the 40 symmetric profiles from TAC$\downarrow_1$. With over 70% of the profiles sampled, we have a reasonably complete description of the 2-player game, TAC$\downarrow_2$, among our 40 strategies. We can identify PSNE simply by examining each strategy pair (s, s'), and verifying whether each is a best response to the other. In doing so, we must account for the fact that our sample data may not include evaluations for all possible profiles.

Definition 6.1 *Profiles can be classified into four disjoint categories, defined below for the 2-player pure-strategy case. (The generalization to n players is straightforward.)*

1. *If (s, s') has not been empirically evaluated, then the estimated expected payoff $\hat{u}(s, s')$ is undefined, and we say (s, s') is* unevaluated*.*
2. *Otherwise, and for some t, $\hat{u}(t, s') > \hat{u}(s, s')$ or $\hat{u}(t, s) > \hat{u}(s', s)$. In this case, we say (s, s') is* refuted*.*
3. *Otherwise, and for some t, (t, s') is unevaluated or (s, t) is unevaluated. In this case, we say (s, s') is a* candidate*.*
4. *Otherwise, in which case we say (s, s') is* confirmed*.*

Based on our TAC$\downarrow_2$ simulations, we have refuted all 586 profiles for which we have estimated payoffs. The remaining 234 are unevaluated.

The definitions above say nothing about the statistical strength of our confirmation or refutation of equilibria. For any particular comparison, one can perform a statistical analysis to evaluate the weight of evidence for or against stability of a given profile. For example, we can construct figures of the form of Figure 6.4, but representing the payoff in response to a particular strategy, rather than in self-play. Figure 6.5 shows such an example, giving the evaluated responses to strategy 34 in the 2-player game. Strategy 16 is the best response in this case. When half the players deviate to strategy 16 from the all-34 profile, they improve their scores by 113 points. We can say this is an improvement, however, only at a $p = 0.14$ significance level. Considering the responses to 16, strategy 5 is best, but insignificantly ($p = 0.47$) better than the next best response, 34. Since 16 is itself a best response to 34, the refutation of (16,34) as a PSNE is statistically weak.

We can also measure the *degree* of refutation in terms of the ε measure (Definition 3.2). Since the payoff function is only partially evaluated, for any profile we have a *lower bound* on ε based on the deviation profiles thus far evaluated. We can generalize the classifications above (refuted, candidate, confirmed) in the obvious way to hold with respect to any given ε level. For example, profile (16,34) is a candidate at $\varepsilon = 5$, but all other profiles are refuted at $\varepsilon > 7$. Based on the above analysis of (34,34) we have that $\varepsilon_{\mathrm{TAC}\downarrow_2}(34, 34) \geq 113$. Figure 6.6 presents the distribution of ε levels at which the 586 evaluated 2-player profiles have been refuted. For example, over half have

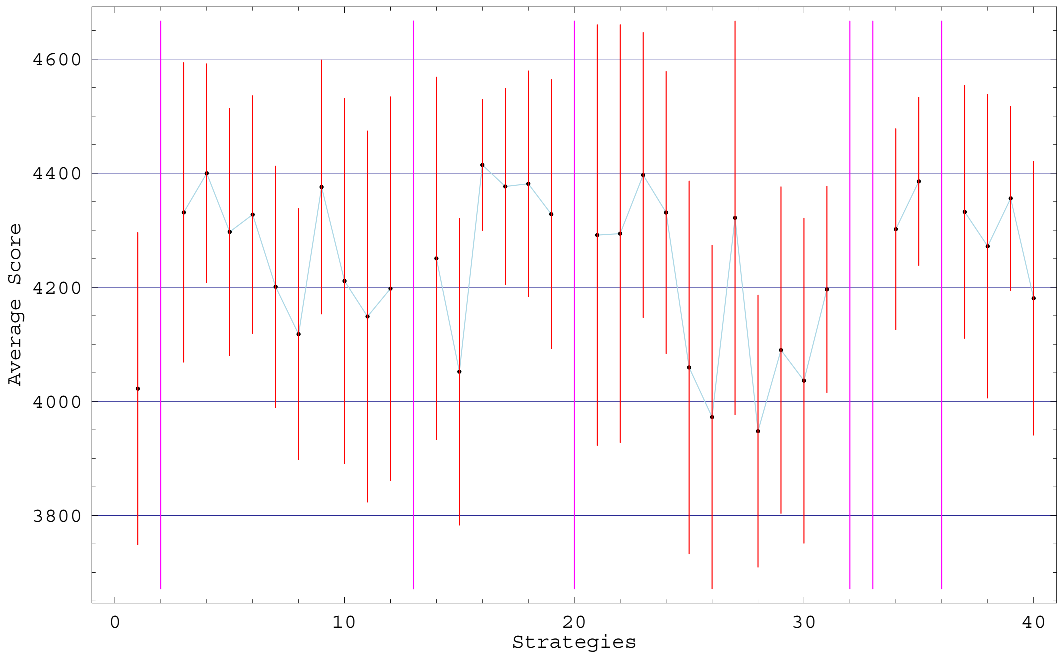

Figure 6.5: Responses to strategy 34 in TAC$\downarrow_2$.

been refuted at $\varepsilon > 258$, and all but ten at $\varepsilon > 31$. These ten pure profiles remain candidates (four of them confirmed) at $\varepsilon = 27$.

We can also evaluate symmetric profiles by considering mixtures of strategies. Although we do not have the full payoff function, we can derive ε bounds on mixed profiles, as long as we have evaluated pure profiles corresponding to all combinations of strategies supported in the mixture. For example, we can derive such bounds for all 546 *pairs* of strategies for which we have evaluated 2-player profiles. The distribution of bounds for these pairs are also plotted in Figure 6.6. Note that the ε bound for a strategy pair is based on the *best* mixture possible of that pair, and so the refutation levels tend to be smaller than for pure strategies. Indeed, the strategy pair (4,9) participates in confirmed symmetric equilibrium, playing strategy 4 with probability 0.729. Another strategy pair—(16,34)—is a candidate at $\varepsilon = 1$, and a total of 11 pairs remain candidates at $eps = 10$, with five confirmed at that level.

As for the SAA experiments in Chapter 5, we apply the term k*-clique* to a set of k strategies such that all profiles involving these strategies are evaluated. A clique defines a subgame of the original game, which can be evaluated by standard methods. We applied iterative elimination of dominated strategies to all the maximal cliques of the 2-player game, ranging in size up to $k = 25$. This indeed pruned many strategies and induced new subsumption relations among the cliques, leaving us with only three maximal cliques—of sizes 16–18. We applied the Lemke-Howson algorithm to these subgames, which identified 29 candidate symmetric equilibria (not refuted by strategies outside the cliques), with distinct supports ranging in size from two to nine. Nineteen of these mixtures are confirmed (including the three pairs mentioned above).

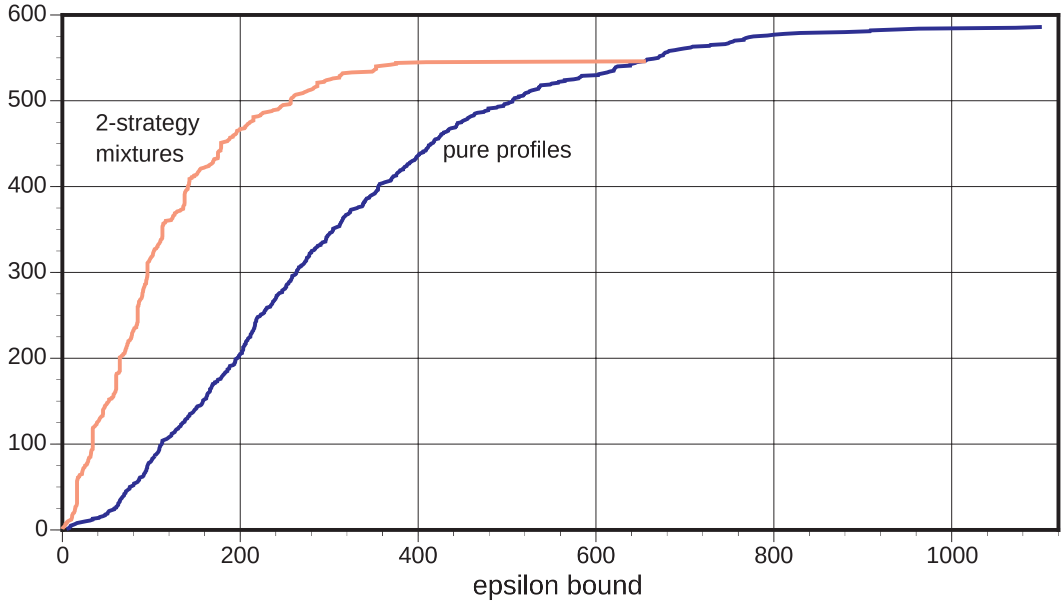

Figure 6.6: Cumulative distribution of ε bounds in TAC$\downarrow_2$.

Because any equilibrium of the full game must also be an equilibrium in any subgame encompassing its support, this exercise also allows us to prune broad regions of profile space from consideration.[5] For instance, the subgame results effectively refute 3056 strategy triples (out of 6545 total, or 47%) as comprising support for symmetric equilibria. By similar reasoning, we refute 14789 strategy quadruples (28%). Given the importance of small supports in recent approaches to deriving equilibria [Porter *et al.*, 2004], or approximate equilibria [Lipton *et al.*, 2003], focusing the search in these regions can be quite helpful.

Finally, we can account for statistical variation in the estimated payoffs by employing sensitivity analysis in our ε calculations (Section 3.8). Naturally, even our confirmed equilibria are refuted with substantial probability, and thus have positive ε in expectation. Although generated from an earlier version of our payoff matrix (fewer simulations and fewer strategies explored), Figure 3.10 on page 44 depicts a typical distribution for ε of a "confirmed" equilibrium profile in TAC$\downarrow_2$.

4-Player Game

Our analysis of the 4-player game, TAC$\downarrow_4$, parallels that of the 2-player game, though of course based on a sparser coverage of the profile space. There are 123,410 distinct TAC$\downarrow_4$ profiles, out of which we have evaluated 2114. Of these, 232 are TAC$\downarrow_2$ profiles with no evaluated neighbors in TAC$\downarrow_4$ (i.e., no deviations tested). Although these are technically PSNE candidates, we distinguish these from PSNE candidates that have actually survived some challenge. The remaining 1876 evaluated profiles are refuted, at various levels. The distribution of ε bounds is plotted in Figure 6.7.

[5] Pruning is strictly justified only under the assumption that we have identified *all* symmetric equilibria of the clique subgames. The Lemke-Howson algorithm does not guarantee this, but in every case for which we were able to check using more exhaustive methods available in GAMBIT, in fact all such equilibria were found.

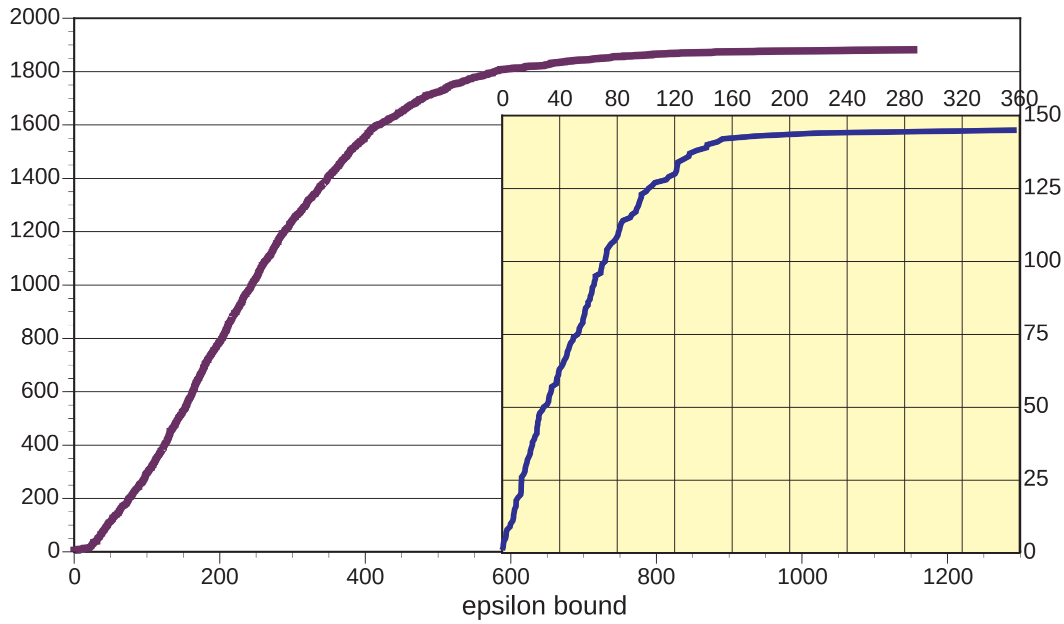

Figure 6.7: Cumulative distribution of ε bounds in TAC $\downarrow_4$. Main graph: pure profiles. Inset: 2-strategy mixtures.

Figure 6.7 also shows, inset, the distribution of ε bounds over the 145 strategy pairs for which we have evaluated all combinations in TAC$\downarrow_4$ (i.e., the 2-cliques). Among these are one candidate equilibrium, and another seven at $\varepsilon = 5$—two of them nearly confirmed. The TAC$\downarrow_4$ cliques are relatively small: three 5-cliques, 16 4-cliques, and 59 3-cliques. Eliminating dominated strategies prunes little in this case, and we have been unsuccesful in getting GAMBIT to solve any k-clique games in the 4-player game for $k > 2$. However, applying replicator dynamics (Section 3.6) produces sample symmetric subgame equilibria, including several mixture triples that constitute candidates with respect to the full game.

Finally, given data in both the 2-player and 4-player games, we can perform some comparisons along the lines of our GAMUT experiments described in Section 3.5. The results, shown in Figure 6.8, are not as clear as those from the known-game experiments, in part because there is no "gold standard", as the 4-player game is quite incompletely evaluated and the 8-player game not at all.

Discussion

Analysis of the various reduced games validates the importance of strategic interactions in TAC. As noted above, the best strategy in self-play (34) is not nearly a best response in most other environments, though it does appear in a few mixed-strategy equilibria of TAC$\downarrow_2$. The strategic situation with strategy 34 is similar to a Prisoners' Dilemma, with 34 the "cooperate" strategy, bidding very low prices for hotels. Defecting against such a strategy means bidding high and exploiting the low prices at the expense of the low bidders. In fact, pairing 34 with most of the other Walverine variants

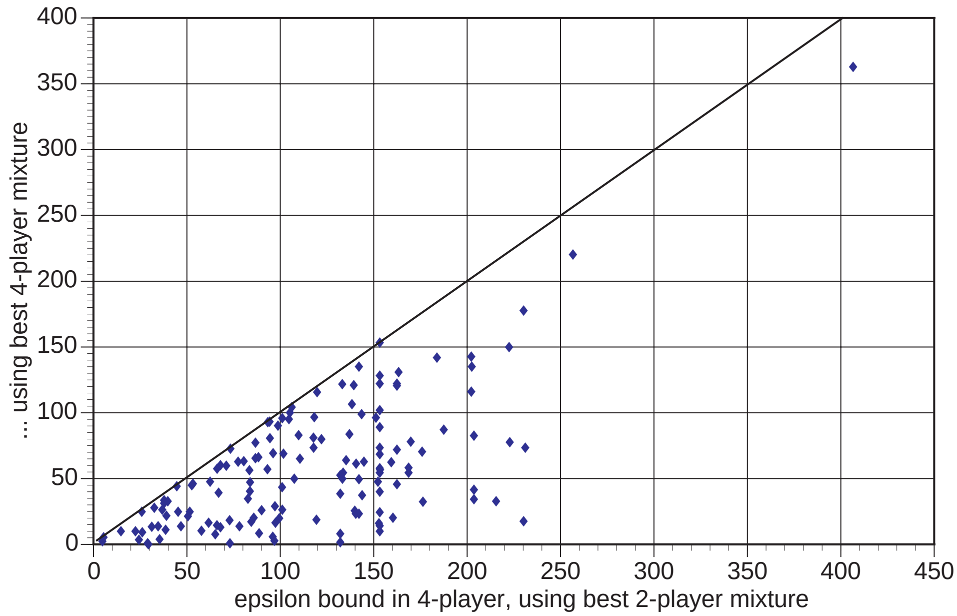

Figure 6.8: ε bounds in the 4-player game achieved by playing the best mixture from the 2-player game, versus playing the best in the 4-player. All points must be southeast of the diagonal by definition.

	17 (D)	**34 (C)**
17 (D)	3971	4377
34 (C)	3907	4302

Table 6.5: The TAC game restricted to strategies 17 and 34 constitutes a Prisoners' Dilemma where 34 is "cooperate" and 17 is "defect". Note that the temptation payoff $>$ reward payoff $>$ punishment payoff $>$ sucker payoff. Additionally, the reward payoff exceeds the average of the temptation and sucker payoffs.

results in a game that is isomorphic to a Prisoners' Dilemma. This is depicted in Table 6.5 with the "defect" strategy being 17 (corresponding to Walverine-04).

Strategy 34 achieves a payoff of 4302 in self-play. For comparison:

- The top scorer in the 2004 tournament, Whitebear, averaged 4122 (adjusted to 4210).
- The best payoff we have found (so far) in TAC$\downarrow_2$ in a two-action mixed-strategy equilibrium candidate ($\varepsilon < 1$) is 4014 (and this involves playing 34 with probability 0.12).
- The best corresponding equilibrium payoff we have found in TAC$\downarrow_4$ is 4018. No such equilibrium includes 34.

6.5 Finding Walverine 2005

The key question remaining after the preceding search in Walverine's parameter space and game-theoretic analysis is: how does it inform our choice of what to play in the tournament? To do this we proceed in two stages:

1. Identify promising strategies in reduced games with Walverine variants.

2. Test these strategies in preliminary rounds of the tournament.

Straightforward solution of the empirical game does not yield a definitive strategy recommendation since there are generally many equilibria. We do have strong evidence that all but a fraction of the original 40 strategies will turn out to be unstable within this set. The supports of candidate equilibria tend to concentrate on a fraction of the strategies, suggesting we may limit consideration to this group. Thus, we employ the preceding analysis primarily to identify promising strategies, and then refine this set through further evaluation in preliminary rounds of the actual TAC tournament.

For the first stage—identifying promising strategies—the 1-player game is of little use. Even discounting strategy 34 (the best strategy in the 1-player game, specially crafted to do well with copies of itself) our experience suggests that strategic interaction is too fundamental to TAC for performance in the 1-player game to correlate more than loosely with performance in the unreduced game. The 4-player game accounts for strategic interaction at a fine granularity, being sensitive to deviations by as few as two of the eight agents. The 2-player game could well lead us astray in this respect. For example, that strategy 34 appears in mixed-strategy equilibria in the 2-player game is likely an artifact of the coarse granularity of that approximation to TAC.

Cooperative strategies like 34 might well survive when deviations consist of half the players in the game, but in the unreduced game we would expect them to be far less stable, i.e., not appearing in any equilibrium. Nonetheless, the correlation between the 2- and 4-player game is high. Furthermore, we have a much more complete description of the 2-player game, with more statistically meaningful estimates of payoffs. Finally, empirical payoff matrices for the 2-player game are far more amenable to our solution techniques, in particular, exhaustive enumeration of symmetric (mixed) equilibria by GAMBIT. For all of these reasons, we focus on the 2-player game for choosing our final Walverine strategies, augmenting our selections with strategies that appear promising in TAC$\downarrow_4$.

Informally, our criteria for picking strong strategies include presence in many equilibria and how strongly the strategy is supported. We start with an exhaustive list of all symmetric equilibria in all cliques of TAC$\downarrow_2$, filtered to exclude any profiles that are refuted in the full game (considering all strategies, not just those in the cliques). There are 68 of these. We next operationalize our criteria for promising strategies with three metrics that we can use to rank strategies given an exhaustive list of equilibria in all cliques of the 2-player game:

- Number of equilibria that the strategy is supported in.

- Maximum mixture probability with which the strategy appears.

- Sum of mixture probabilities across all equilibria.

Applying these metrics to the set of equilibria yields the strategy ranking in Table 6.6.

Based on this analysis, we chose $\{4, 16, 17, 35\}$ as the most promising candidates, and added $\{3, 37, 39, 40\}$ based on their promise in TAC$\downarrow_4$. Figure 6.9 reveals strategies 37 and 40 to be the

Strategy	Count	Max	Sum
17	24	0.499	5.39
4	19	0.729	4.86
21	17	0.501	4.68
16	18	0.892	3.77
23	14	0.542	3.34
6	16	0.699	3.25
9	16	0.367	3.09
5	23	0.247	2.63
24	17	0.232	2.34
35	15	0.641	2.04
40	20	0.180	1.95
3	8	0.401	1.70
34	7	0.307	1.05
7	6	0.099	0.37
38	5	0.091	0.35
39	3	0.126	0.18

Table 6.6: For all strategies appearing in an unrefuted equilibrium of a clique in TAC$\downarrow_2$, the number of equilibria, the maximum mixture probability, and the sum of all mixture probabilities across equilibria.

top two candidates after the seeding rounds. In the semifinals we played 37 and 40 and found that 37 outperformed 40, 4182 to 3945 ($p = .05$). Based on this, we played 37 as the Walverine strategy for the finals in TAC-05.

6.6 TAC 2005 Outcome

Officially, Walverine placed third, based on the 80 games of the 2005 finals. However, it was clear to all present that had it not been for a network glitch that left Walverine unable to play in two games, we would have come out in first. In fact, 22 games were tainted when RoxyBot (shown in Table 6.7 in second place) had a serious malfunction[6] that dropped it to last place. Since games with erratic agent behavior add noise to the scores, the TAC organizers published semi-official results with the errant RoxyBot games removed. Walverine's missed games occurred during those games and so Walverine was (semi-officially) the top-scoring agent (Table 6.7). Figure 6.10 shows the adjusted scores with error bars. Walverine beat the runner up (RoxyBot) at the $p = 0.17$ significance level. Regardless of the ambiguity (statistical or otherwise) of Walverine's victory in the competition, we consider its strong performance under real tournament conditions to be evidence (albeit limited) of the efficacy of our approach to strategy generation in complex games such as TAC.

[6]The malfunction was actually a simple human error: instead of playing a copy of the agent on each of the two game servers per the tournament protocol, the RoxyBot team accidentally set both copies of the agent to play on the same server and none on the other. RoxyBot not only failed to participate in the first server's games, but placed double bids in games on the other server.

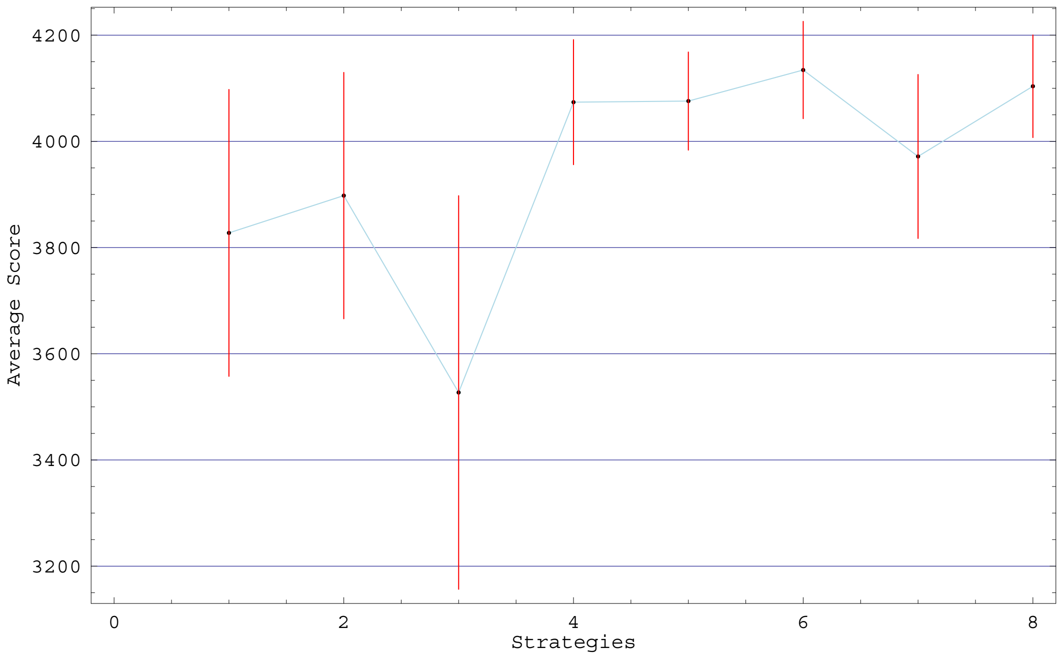

Figure 6.9: Performance of eight Walverine variants, $\{3, 4, 16, 17, 35, 37, 39, 40\}$, in the TAC-05 seeding rounds, based on 507 games.

Agent	Raw Score	Adjusted Score	95% C.I.
Walverine	4157	4132	± 138
RoxyBot	4067	4030	± 167
Mertacor	4063	3974	± 152
whitebear05	4002	3902	± 130
Dolphin	3993	3899	± 149
SICS02	3905	3843	± 141
LearnAgents	3785	3719	± 280
e-Agent	3367	3342	± 117

Table 6.7: Scores, adjusted scores, and 95% mean confidence intervals on control variate adjusted scores for the 58 games of the TAC Travel 2005 finals, after removing the first 22 tainted games. (LearnAgents experienced network problems for a few games, accounting for their high variance and lowering their score.)

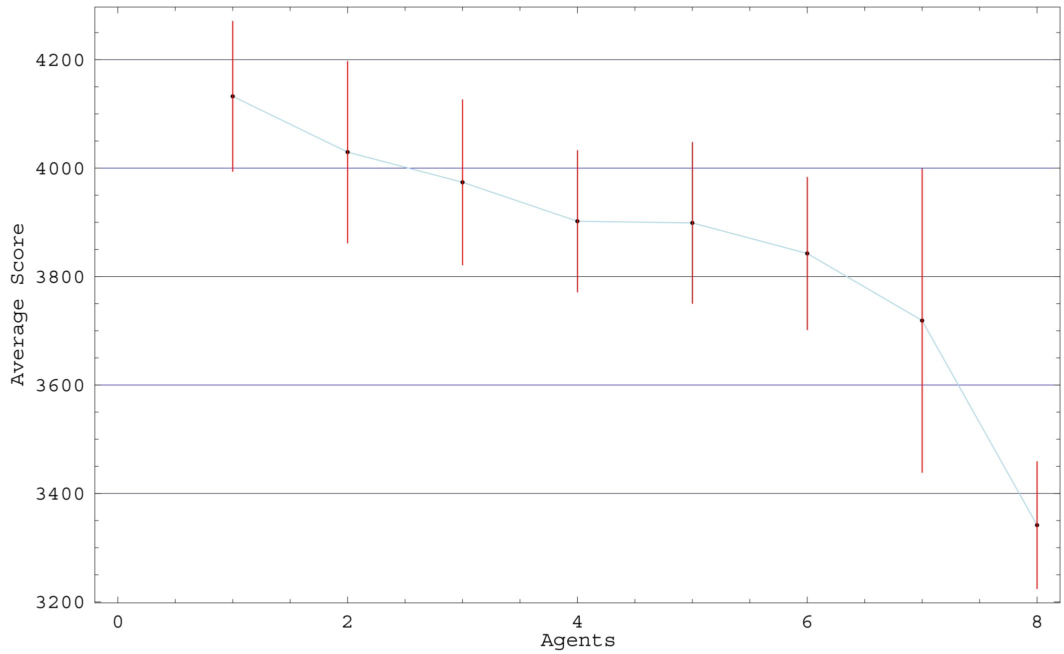

Figure 6.10: Comparison of the eight agents in the TAC-05 finals.

6.7 Related Work: Strategic Interactions in TAC Travel

That strategic choices interact, and implications for design and evaluation, have been frequently noted in the TAC literature. In a report on the first TAC tournament, Greenwald and Stone [2001] observe that the strategy of bidding high prices for hotels performed reasonably in preliminary rounds, but poorly in the finals when more agents were high bidders (thus raising final prices to unprofitable levels). Stone *et al.* [2001] evaluate their agent ATTac-2000 in controlled post-tournament experiments, measuring relative scores in a range of contexts, varying the number of other agents playing high- and low-bidding strategies. A report on the 2001 competition [Wellman *et al.*, 2003b] concludes that the top scorer, LivingAgents, would perform quite poorly against copies of itself. The designers of SouthamptonTAC [He and Jennings, 2002] observed the sensitivity of their agent's TAC-01 performance to the tendency of other agents to buy flights in advance, and redesigned their agent for TAC-02 to attempt to classify the competitive environment faced and adapt accordingly [He and Jennings, 2003]. ATTac-2001 explicitly took into account the identity of other agents in training its price-prediction module [Stone *et al.*, 2003]. To evaluate alternative learning mechanisms through post-competition analysis, Stone et al. recognized the effect of the policies on the outcomes being learned, and thus adopted a carefully phased experimental design in order to account for such effects.

One issue considered by several TAC teams is how to bid for hotels based on predicted prices and marginal utility. Greenwald and Boyan [2004] have studied this in depth, performing pairwise comparisons of four strategies, in profiles with four copies of each agent.[7] Their results indicate

[7] By the terminology in this thesis, their trials focused on the 2-player reduced version of the game.

that absolute performance of a strategy indeed depends on what the other agent plays.

By far the most extensive experimental analysis reported for TAC Travel to date is that performed by Vetsikas and Selman [2003]. In the process of designing Whitebear for TAC-02, they first identified candidate policies for separate elements of the agents' overall strategy. They then defined extreme (boundary) and intermediate values for these partial strategies, and performed experiments according to a systematic and deliberately considered methodology. Specifically, for each run, they fix a particular number of agents playing intermediate strategies, varying the mixture of boundary cases across the possible range. In all, the Whitebear experiments comprised 4500 profiles, with varying *even* numbers of candidate strategies (i.e., profiles of the 4-player game). Their design was further informed by 2000 games in the preliminary tournament rounds. This systematic exploration was apparently helpful, as Whitebear was the top scorer in the 2002 tournament. This agent's predecessor version placed third in TAC-01, following a less comprehensive and structured experimentation process. Its successor placed third again in 2003, and regained its first-place standing in 2004. Since the rules were adjusted for TAC-04, this outcome required a new regimen of experiments. Most recently, Whitebear placed second in 2005.

Chapter 7

Conclusion

IN WHICH *we review our strategy generation techniques and consider the holy grail of a general strategy generation engine.*

This thesis presents new approaches to strategy generation and applies them to a variety of games. The key contributions are the following:

1. An algorithm to compute best-response strategies in a class of 2-player, one-shot, infinite games of incomplete information.

2. An empirical game methodology for applying game-theoretic analysis to much larger games than previously possible.

3. Theoretical and experimental evidence of the efficacy of our methodology.

4. A price-prediction approach to strategy generation in simultaneous auctions for complementary goods.

5. Application of the above methods to find good strategies in complex games.

7.1 Summary of Contribution

In Chapter 2 I have presented a proof that best responses to piecewise linear strategies in a class of infinite games of incomplete information are piecewise linear. The proof is constructive and contains a polynomial-time algorithm for finding such best responses. To my knowledge, this is the first algorithm for finding best-response strategies in a broad class of infinite games of incomplete information.

For some games, this best-response algorithm can be iterated to find Bayes-Nash equilibria. It remains a goal to characterize the class of games for which iterated best response converges. Our method confirms known equilibria from the literature (e.g., auction games such as FPSB, Vickrey, and bargaining), confirms an equilibrium I derive here (in the Supply Chain game), and discovers equilibria in new games (the Shared Good auction and the Joint Purchase auction).

Chapter 3 presents our empirical game methodology. I am not the first to present the core idea of estimating a restricted form of a complex game via Monte Carlo simulation (see Section 3.9) but this thesis contributes several new innovations. Most notable is the approach of player reduction (Section 3.5) for which I demonstrate the potential for vast computational savings. Furthermore, I present theoretical and empirical results establishing its soundness as an approximation method in certain games. Also in Chapter 3 I describe standard variance reduction techniques, showing how to frame game simulation to be amenable to their application. I present results establishing the value of control variates applied to the First-Price Sealed-Bid Auction game (FPSB).

Next in Chapter 3 I consider the problem of finding equilibria in empirically estimated payoff matrices. To some extent this is addressed by off-the-shelf solvers (notably GAMBIT). For this problem, the key contribution in this thesis is the application of replicator dynamics to finding equilibria in large games. Elsewhere in the literature, replicator dynamics is, to my knowledge, treated exclusively in terms of the equilibrium selection problem. I discuss game solution methods in detail in Section 3.6. Finally, given an equilibrium in an empirical game, I present new methods for assessing the quality of the solution with respect to the full game by defining a belief distribution over payoff matrices for the underlying game (Section 3.8).

Chapter 4 lays out a general approach to generating candidate strategies in games involving bidding for complementary goods in simultaneous auctions: price prediction. In this chapter I describe two games—the Trading Agent Competition (TAC) travel-shopping game and the Simultaneous Ascending Auctions (SAA) mechanism—and present families of strategies that are applicable in both. Borne out in the studies in the remaining chapters is the conjecture that price prediction is a fundamental strategic issue in both TAC and SAA. I describe the key concept of distribution prediction in these domains and consider various ways to generate price distributions. In particular, I consider (1) the model-based approach of Walrasian price equilibrium prediction and (2) self-confirming predictions, a prediction strategy defined such that in equilibrium it will be correct. Additionally, I describe a new performance-based measure of prediction quality and discuss the results of its application in TAC. With both the performance-based measure and an absolute accuracy measure (Euclidean distance from actual prices) I demonstrate the power of price prediction based on economic models. Even more compelling, using self-confirming distributions, prediction accuracy is by definition probabilistically perfect.

Chapter 5 presents the application of our empirical game methodology to the SAA game. I first describe the type distributions we employ, defining a set of SAA environments. I then consider three families of strategies in turn, performing restricted game analysis in each for various environments. For strategies parameterized by sunk-awareness (Sections 4.4 and 5.2) we find that good settings of this parameter depend sensitively on the SAA environment but that the equilibrium parameter setting varies predictably with the number of agents in the game. I next consider basic price predictors, demonstrating a marked improvement over our benchmark strategy, straightforward bidding (SB). I also analyze the nature of the performance improvement and the effect on market efficiency. Finally, I consider self-confirming distribution prediction and show that it is a good strategy (based on the ε-equilibrium measure) in a range of SAA environments.

Chapter 6 parallels Chapter 5 in applying Chapter 3's methodology to a family of strategies defined in Chapter 4. In particular, I present parametric variants of our TAC entrant, the Walrasian price predictor, Walverine. After a massive year-long simulation effort, we have estimated a partial payoff matrix for the TAC game from which we have been able to gain sufficient strategic insight to help choose our strategy for the 2005 competition, with huge success.

7.2 Future Work

A key question which I have only begun to address is: how well and under what circumstances will our methods succeed? The approaches in this thesis can approximate an equilibrium in the case that the restricted strategy space contains that equilibrium. For many simple games this is not hard to ensure. It is then also often possible to confirm the equilibrium of the restricted game in the full (or less restricted) game. For games of the complexity of TAC, although our methods represent the state of the art, we have no reason to believe that our set of candidate strategies contains a true equilibrium. Although our chosen strategy performs well (arguably the most competent strategy in the 2005 competition) it is difficult to conclude much about our empirical methodology in such complex domains. Future work should more thoroughly characterize the class of games to which this empirical game-approximation approach is amenable.

The ultimate goal of a Strategy Generation Engine is to take as input only the description of a game (strategy sets, type distributions, and payoff function) and output Nash equilibrium strategy profiles. Thus, another key area for future work is to take further steps toward removing the manual steps in our current process. These include automation of the following:

1. Selection of, or search for, seed strategies for the best-response solver for infinite games, when iterated best response does not converge from a default strategy such as truthful bidding.[1]
2. Generation of candidate strategies from which to generate empirical payoff matrices, in games with strategy spaces more complex than one-dimensional functions from type to action.
3. Choice of profiles to sample when only a partial payoff matrix is possible (as in TAC).

Whether or not we succeed in all these areas, a next natural question is: What can one do with a Strategy Generation Engine? Beyond the obvious application for participants of games, our methods can be instrumental for mechanism designers. As mechanism designers we set the rules of a game in order to achieve various objectives like fairness and aggregate utility (i.e., market efficiency). But to assess such outcomes, we must know what the agents will do in the face of alternative game rules. This is precisely what a strategy generator (game solver) tells us. And, as noted in Chapter 1, it often suffices to have a strategy generator that returns a single Nash equilibrium (even when there are many) as there are good reasons to believe such an equilibrium predicts actual agent play.

In fact, this argument can be strengthened when we have control over the mechanism. A mechanism is *incentive compatible* if each agent's equilibrium strategy is to simply submit its type (e.g., bid truthfully). Following the constructive proof of the revelation principle [Mas-Colell *et al.*, 1995], any mechanism can be made incentive compatible by finding some equilibrium profile and then wrapping the original mechanism in a new interface: the allowable actions are now the possible types and the wrapper takes the submitted types, feeding them to the equilibrium strategies to produce actions for the original game. This shows that as long as other agents play a truth-telling strategy (submit their types) an agent can do no better than to do so as well. Thus, not only is a strategy generator instrumental in implementing a direct revelation mechanism, doing so helps resolve the equilibrium selection problem.

Is the holy grail of a Strategy Generation Engine attainable? A fully general and fully automated game solver is out of reach for the foreseeable future. Nonetheless, for broad classes of games, this thesis presents methods for strategy generation and analysis, generates strategies for specific games, and establishes them as highly successful.

[1] Also promising are alternatives to iterated best response such as fictitious play.

Appendix A

Proofs and Derivations

A.1 CDF of a Piecewise Uniform Distribution

Given a piecewise uniform distribution with pdf

$$f(x) = \begin{cases} 0 & \text{if } -\infty < x \leq d_2 \\ f_2 & \text{if } d_2 < x \leq d_3 \\ \dots & \\ f_{J-1} & \text{if } d_{J-1} < x \leq d_J \\ 0 & \text{if } d_J < x \leq +\infty \end{cases}$$

it is straightforward to derive the cdf (defining $f_1 \equiv 0$, $f_J \equiv 0$, $d_1 \equiv -\infty$, and $d_{J+1} \equiv +\infty$):

$$F(x) = \begin{cases} f_i \cdot (x - d_i) + \sum_{j=1}^{i-1} f_j \cdot (d_{j+1} - d_j) & \text{if } d_i < x \leq d_{i+1}. \end{cases}$$

The expectation (not needed for our best-response algorithm) is

$$\sum_{i=0}^{n-1} \frac{f_i}{2} (d_{i+1}^2 - d_i^2).$$

A.2 Proof of Theorem 2.3

We show that the following is a symmetric Bayes-Nash equilibrium when $v \in [3/2, 3]$ (which includes the game with $v = 1.55$ analyzed in Section 2.4):

$$s(t) = \begin{cases} 2v/3 - 1/2 & \text{if } t \leq 2v/3 - 1 \\ t/2 + v/3 & \text{otherwise.} \end{cases} \tag{A.1}$$

The expected utility (surplus) for agent type t bidding a against agent type $T \sim U[0,1]$ playing

(A.1) is:

$$\begin{aligned} EU(t,a) &= E_T\left[u(t,a,T,s(T)\right] \\ &= (a-t)\Pr(a+s(T)\leq v) \\ &= (a-t)\Big[\Pr(T\leq 2v/3-1) \\ &\quad \cdot\Pr(a+2v/3-1/2\leq v \mid T\leq 2v/3-1) \\ &\quad +\Pr(T>2v/3-1) \\ &\quad \cdot\Pr(a+T/2+v/3\leq v \mid T>2v/3-1)\Big]. \end{aligned}$$

We consider three cases, conditioned on the range of a.

Case 1: $a \leq 2v/3-1/2$.

The expected surplus is:

$$\begin{aligned} EU(t,a) &= (a-t)[(2v/3-1)+(2-2v/3)] \\ &= a-t. \end{aligned}$$

Since the expected surplus is monotonic in a, the optimal bid is found at the upper boundary, namely, $a = 2v/3-1/2$, giving us $EU(t,a) = 2v/3-1/2-t$.

Case 2: $a \in [2v/3-1/2,\ v/3+1/2]$.

The expected surplus is:

$$\begin{aligned} EU(t,a) =&(a-t)[(2v/3-1)+(2v/3-2a+1)] \\ =&(a-t)(4v/3-2a). \end{aligned} \tag{A.2}$$

Equation A.2 is maximized at $a = t/2+v/3$. With $v \in [3/2\ ,3]$, we need to consider whether this point occurs below the lower boundary of a. This gives us two cases. If $t > 2v/3-1$ then the maximum of (A.2) lies in the range of bids we consider here, and the best response is $a_1 = t_1/2+v/3$, giving us $EU(t,a) = (3t-2v)^2/18$. If $t > 2v/3-1$ then the maximum occurs at the lower boundary, and the best response is $a = 2v/3-1/2$, giving us $EU(t,a) = 2v/3-1/2-t$.

Case 3: $a > v/3+1/2$.

The expected surplus is always zero in this range.

The expected surplus of Case 2 is always positive, and always at least as high as Case 1, hence it must specify the best-response policy. But Case 2 gives us the bidding policy specified by (A.1) when $v \leq 3$.

We have shown that (A.1) is a best response to itself for all $v \in [3/2, 3]$, hence it is a symmetric Bayes-Nash equilibrium. □

A.3 Proof of Theorem 2.4

We show that the following is a Bayes-Nash equilibrium of the auction game from Section 2.4:

$$a(t) = \frac{2t+A}{3}. \tag{A.3}$$

Given that the other agent employs the strategy in (A.3) for its type $T \sim U[0,1]$, we show that (A.3) is a best response. The expected utility (surplus) for an agent playing bid a against an agent bidding according to (A.3) is

$$\begin{aligned} EU(t,a) &= E\left[u\left(t, a, \frac{2T+A}{3}\right)\right] \\ &= (t-a/2)\Pr(a > s(T)) + E\left[\frac{1}{2}\cdot\frac{2T+A}{3} \;\middle|\; a < s(T)\right]\Pr(a < s(T)) \\ &= (t-a/2)\Pr\left(\frac{3a-A}{2} > T\right) \\ &\quad + E\left[\frac{2T+A}{6} \;\middle|\; \frac{3a-A}{2} < T\right]\cdot\Pr\left(\frac{3a-A}{2} < T\right). \end{aligned} \tag{A.4}$$

We consider two cases, conditioned on the range of a.

Case 1: $a > \dfrac{2B+A}{3}$.
The expected surplus is:

$$EU(t,a) = t - a/2$$

which implies an optimal action at the left boundary:

$$a_1^* = \frac{2B+A}{3}.$$

Case 2: $a < \dfrac{2B+A}{3}$.
In this case, both the probabilities in (A.4) are nonzero and the expected surplus is

$$EU(a) = (t-a/2)\frac{\frac{3a-A}{2} - A}{B-A} + 1/6\left(2\cdot\frac{\frac{3a-A}{2}+B}{2} + A\right)\cdot\frac{B - \frac{3a-A}{2}}{B-A},$$

which implies an optimal action where the derivative with respect to a is zero:

$$a_2^* = \frac{2t+A}{3}.$$

Comparing the expected surpluses of the candidate action functions,

$$EU(a_2^*) - EU(a_1^*) = \frac{(B-t)^2}{2(B-A)} > 0.$$

Therefore a_2^* is a best response to (A.3), i.e., itself, and therefore a Bayes-Nash equilibrium. □

A.4 Proof of Theorem 2.5

We show that the following is a Bayes-Nash equilibrium of the Vicious Vickrey auction described in Section 2.4:

$$a(t) = \frac{k+t}{k+1}. \tag{A.5}$$

Given that the other agent bids according to (A.5) for its type $T \sim U[0,1]$, we show that (A.5) is a best response. The expected utility for bidding a against an agent playing (A.5) is

$$\begin{aligned} EU(t,a) &= E\left[-k(T-a) \;\middle|\; a < \frac{T+k}{1+k}\right] \cdot \Pr\left(a < \frac{T+k}{1+k}\right) \\ &\quad + E\left[(1-k)\left(t - \frac{T+k}{1+k}\right) \;\middle|\; a > \frac{T+k}{1+k}\right] \cdot \Pr\left(a > \frac{T+k}{1+k}\right) \\ &= -k\left(\frac{1+a(1+k)-k}{2} - a\right)(1 - a(1+k)+k) \\ &\quad + (1-k)\left(t - \frac{1/2 \cdot a(1+k) - 1/2 \cdot k + k}{1+k}\right)(a(1+k)-k). \end{aligned} \tag{A.6}$$

Tie-breaking cases can be ignored here since they occur with zero probability given the type distribution and form of the opponent strategy. We can now simply check the first-order condition of (A.6) to find the maximizing a, which is the best response:

$$a^* = \frac{k+t}{k+1}.$$

Since (A.5) is a best response to itself, it is a symmetric Bayes-Nash equilibrium. □

A.5 Proof of Theorem 3.4

We derive the expected payoff for an agent playing strategy $k_i t$ against everyone else playing kt in FPSBn (cf. Definition 3.3). (We denote agent j's type by $T_j \sim U[0,1]$ and collectively by the vector $\boldsymbol{T}$.)

$$u_i(k_i,k) = E_{\boldsymbol{T}}\left[T_i - k_i T_i \;\middle|\; k_i T_i > \max_{j \neq i}(kT_j)\right] \cdot \Pr\left(k_i T_i > \max_{j \neq i}(kT_j)\right). \tag{A.7}$$

If $k_i = k = 0$ then all players bid zero and each agent's expected payoff is its expected type ($1/2$) times its probability of being randomly selected as the winner ($1/n$). That is,

$$u_i(k_i,k) = \frac{1}{2n} \quad \text{if } k_i = k = 0. \tag{A.8}$$

Otherwise, the probability in (A.7) can be rewritten as

$$\Pr\left(\frac{\max_{j \neq i}(T_j)}{T_i} < \frac{k_i}{k}\right).$$

Letting $r \equiv k_i/k$ and $Y \equiv max_{j \neq i}(T_j)$, the cdf and pdf of Y—a random variable which is the maximum of $n-1$ random variables—and T_i are

$$F_{T_i}(x) = x \quad \text{and} \quad f_{T_i}(x) = 1 \quad \text{for } x \in [0,1] \quad \text{(i.e., } T_i \sim U[0,1]\text{);}$$
$$F_Y(y) = y^{n-1} \quad \text{and} \quad f_Y(y) = (n-1)y^{n-2} \quad \text{for } y \in [0,1].$$

We now calculate the cdf of $Z \equiv Y/T_i$:

$$\begin{aligned} F_Z(r) &= \int_0^\infty \int_{-\infty}^{xr} f_{T_i}(x) f_Y(y)\, dydx \\ &= \int_0^\infty \int_{-\infty}^{xr} (n-1)y^{n-2}\, dydx. \end{aligned}$$

Since the probability density is only positive for $T_i,\ Y \in [0,1]$ the integration region has two different shapes depending on whether $r < 1$, and so the above integral is $F_Z(r) =$

$$\begin{cases} \displaystyle\int_0^1 \int_0^{xr} (n-1)y^{n-2}\, dydx = \frac{r^{n-1}}{n} & \text{if } r < 1 \\ \displaystyle\int_0^{1/r} \int_0^{xr} (n-1)y^{n-2}\, dydx + \int_{1/r}^1 \int_0^1 (n-1)y^{n-2}\, dydx = \frac{1}{nr} + \frac{r-1}{r} & \text{otherwise.} \end{cases}$$

We can now rewrite (A.7) as

$$(1-k_i)E\left[T_i \mid Z < r\right] \cdot F_Z(r). \tag{A.9}$$

Let X be the random variable T_i conditionalized by $Z < r$, then

$$\begin{aligned} F_X(t) = \Pr(T_i < t \mid Z < r) &= \frac{\Pr(T_i < t,\ Z < r)}{\Pr(Z < r)} = \frac{\Pr(Y/r < T_i < t)}{\Pr(Z < r)} \\ &= \frac{\int_0^t \int_0^{rx} f_Y(y) f_{T_i}(x)\, dydx}{F_Z(r)}. \end{aligned} \tag{A.10}$$

For the case $r < 1$, the double integral in the numerator evaluates to $r^{n-1}t^n/n$ and the denominator to r^{n-1}/n, yielding

$$F_X(t) = t^n \quad \text{and} \quad f_X(t) = nt^{n-1} \quad \text{for } t \in [0,1].$$

And so the expectation in (A.9) is

$$E[T_i \mid Z < r] = \int_0^1 t \cdot nt^{n-1}\, dt = \frac{n}{n+1} \quad \text{(for } r < 1\text{).} \tag{A.11}$$

For the case $r \geq 1$, (A.10) becomes

$$\frac{\int_0^{\min(1/r,t)} \int_0^{rx} (n-1)y^{n-2}\,dydx + \int_{\min(1/r,t)}^{t} \int_0^{1} (n-1)y^{n-2}\,dydx}{\frac{1}{nr} + \frac{r-1}{r}}$$

$$= \frac{nr(t - \min(\frac{1}{r}, t)) + r^n \min(\frac{1}{r}, t)^n}{1 + n(r-1)}.$$

Differentiating yields the pdf,

$$f_X(t) = \begin{cases} 1 + n(rt-1) & \text{if } t > \frac{1}{r} \\ \dfrac{r^n t^n}{1+n(r-1)} & \text{otherwise.} \end{cases}$$

And so the expectation in (A.9) with $r \geq 1$ (equivalently, $k_i \geq k$) is

$$\int_0^{\frac{1}{r}} t \cdot \frac{r^n t^n}{1+n(r-1)}\,dt + \int_{\frac{1}{r}}^{1} t \cdot (1 + n(rt-1))\,dt$$

$$= \frac{n(r^2 + n(r^2-1)+1)}{2r(n+1)(n(r-1)+1)} \quad \text{(for } r \geq 1\text{).} \tag{A.12}$$

Combining (A.8), (A.9), (A.11), and (A.12) and substituting back k_i/k for r yields a closed-form expression for the expected payoff:

$$u_i(k_i, k) = \begin{cases} \dfrac{1}{2n} & \text{if } k_i = k = 0 \\ \dfrac{1-k_i}{n+1} \left(\dfrac{k_i}{k}\right)^{n-1} & \text{if } k_i \leq k \\ \dfrac{(1-k_i)((n-1)k_i^2 - (n-1)k^2)}{2(n+1)k_i^2} & \text{otherwise.} \quad \square \end{cases}$$

A.6 Proof of Theorem 3.5

$\text{BR}(k) \equiv \arg\max_{k_i} u_i(k_i, k)$ is the best response to everyone else playing kt. We find this by finding best-response candidates, i.e., maximizing the pieces of $u_i(k_i, k)$ by setting the derivatives with respect to k_i to zero. Then we compare the candidates to find the overall best response. The best-response candidates for $u_i(k_i, k)$, as found by Mathematica, are:

$$\psi \equiv \frac{n-1}{n}, \qquad \xi \equiv$$
$$\frac{\sqrt[3]{3}\left(k^2\left(n^2-1\right)\left(9n+\sqrt{3(n+1)\left((n-1)k^2+27(n+1)\right)}+9\right)\right)^{2/3}-3^{2/3}k^2\left(n^2-1\right)}{3(n+1)\sqrt[3]{k^2(n-1)\left(9n^2+18n+(n+1)^{3/2}\sqrt{3(n-1)k^2+81(n+1)}+9\right)}}.$$

We first treat the special case of $k = 0$. As in general FPSB with unrestricted strategies, there is no best response to the other agents all bidding zero: the lower your bid the better as long as it's strictly positive. For $k > 0$, the condition $u_i(\psi, k) > u_i(\xi, k)$ reduces to $k \geq \frac{n-1}{n}$ and so the best response to a symmetric profile of strategy k is:

$$\mathrm{BR}(k) = \begin{cases} \text{undefined} & \text{if } k = 0 \\ \xi & \text{if } k < \dfrac{n-1}{n} \\ \dfrac{n-1}{n} & \text{if } k \geq \dfrac{n-1}{n}. \quad \square \end{cases}$$

A.7 Proof of Lemma 3.7

To show that $f(k)$ has no roots in $(0, +\infty)$ except $\frac{n-1}{n}$ we establish a set of points guaranteed to be a superset of the set of roots of $f(k)$ and show that none, other than $\frac{n-1}{n}$, are positive. We first apply the following transformations to f to combine common subexpressions and make all the coefficients integers:[1]

$$(n-1)k^2 + 27(n+1) \to \frac{3a^2}{n+1} \qquad \text{and}$$
$$\left(9n+3a+9\right)\left(n^2-1\right)k^2 \to 9b^3. \tag{A.13}$$

These transformations have no effect as long as

$$a = \sqrt{\frac{(n+1)\left((n-1)k^2+27(n+1)\right)}{3}} \qquad \text{and}$$
$$b = \sqrt[3]{\frac{(3a+9n+9)\left(n^2-1\right)k^2}{9}}, \tag{A.14}$$

or, equivalently, when the following polynomials are zero:

$$(n+1)\left((n-1)k^2+27(n+1)\right) - 3a^2 \qquad \text{and} \tag{A.15}$$

$$(3a+9n+9)\left(n^2-1\right)k^2 - 9b^3. \tag{A.16}$$

[1] This and following key insights that made this proof possible are due to Maxim Rytin. Andrzej Kozlowski and Daniel Lichtblau also provided invaluable guidance.

(In fact, transforming an equation of the form $x = y$ into $x^2 = y^2$ allows for additional spurious solutions such as $x = 1$ and $y = -1$. This does not affect the proof since we need only a guarantee that our set of possible roots is a superset of the actual roots.) After applying (A.13), simplifying (using the fact that n, a, and b are all positive), and collecting the terms over a common denominator, we can write the numerator of $f(k)$ as

$$\begin{aligned}&\left(-3b^2+3k(n+1)b+k^2\left(n^2-1\right)\right)\Big(9b^4+9(k-1)(n-1)b^3\\&\quad-3k(2k+3)\left(n^2-1\right)b^2-3(k-1)k^2(n-1)^2(n+1)b+k^4\left(n^2-1\right)^2\Big)\end{aligned} \tag{A.17}$$

and the denominator as

$$6b(n+1)\left(k^2\left(n^2-1\right)-3b^2\right)^2 \tag{A.18}$$

with a and b defined by (A.14).

We can now find all possible roots of (A.17) given (A.14) by finding the common roots of (A.17), (A.15), and (A.16). To do this, we apply Buchberger's [1982] algorithm to compute a Gröbner basis,[2] resulting in

$$k^8(n-1)^4(n+1)^5(kn-n+1)^2\left(nk^2-k^2+2nk+6k+n-1\right), \tag{A.19}$$

which for $n > 1$ has the following real roots:

$$\left\{-\frac{n+2\sqrt{2}\sqrt{n+1}+3}{n-1},\quad -\frac{n-2\sqrt{2}\sqrt{n+1}+3}{n-1},\quad 0,\quad \frac{n-1}{n}\right\} \tag{A.20}$$

These comprise all the possible real roots of the numerator and so there can be no other roots of $f(k)$. (A root of the numerator is a root of the quotient unless also a root of the denominator.) We verify directly using simplification and reduction routines in Mathematica, that $f(\frac{n-1}{n}) = 0$ for all $n > 1$. This is done by establishing that the numerator of $f(\frac{n-1}{n})$ is equal to zero for all $n > 1$ and, likewise, that the denominator is not.[3] The numerator of $f(\frac{n-1}{n})$ is

$$\begin{aligned}&\sqrt[3]{3}(n-1)^6\left(3^{2/3}n^2-3\nu n-\sqrt[3]{3}\left((n+1)\left(9n^2+2\mu n+9n+\mu\right)\right)^{2/3}-3^{2/3}\right)\\&\cdot\left(-3^{2/3}n^2-3\nu n-3\nu+\sqrt[3]{3}\left((n+1)\left(9n^2+2\mu n+9n+\mu\right)\right)^{2/3}+3^{2/3}\right)^2\end{aligned}$$

where

$$\begin{aligned}\mu &\equiv \sqrt{3(n+1)(7n-1)} \quad \text{and}\\ \nu &\equiv \sqrt[3]{(n+1)\left(9n^2+2\mu n+9n+\mu\right)}.\end{aligned}$$

[2] Gröbner bases [Becker and Weispfenning, 1993] are a generalization of Gaussian elimination to the case of polynomial systems. The set of polynomials G comprising the Gröbner basis of a set of polynomials P has the property that it shares the same set of common roots as P. Buchberger's algorithm is implemented in the Mathematica function `GroebnerBasis`.

[3] The following evaluating to `True` comprises the computer-assisted proof (with thanks yet again to Maxim Rytin):

```
simp = Function[x, Simplify[Together//@x, n>1]];
Reduce[ForAll[n, n>1, simp@simp@Numerator@Together@f[n,(n-1)/n]]]
```

And similarly for the denominator.

The denominator is

$$\left(-\sqrt[3]{3}n^2 + ((n+1)(2n\mu + \mu + 9n(n+1)))^{2/3} + \sqrt[3]{3}\right)^2 18(n-1)^5 n(n+1)^{4/3}$$
$$\cdot \sqrt[3]{2n\mu + \mu + 9n(n+1)}.$$

It remains to show that no other points in (A.20) are positive. The first is clearly negative. To show that the second is also negative it suffices to show that $n + 3 - 2\sqrt{2}\sqrt{n+1} > 0$ which is the case because

$$\begin{aligned} & n > 1 \\ \implies & (n-1)^2 > 0 \\ \implies & n^2 - 2n + 1 > 0 \end{aligned}$$

$$\begin{aligned} \implies & n^2 + 6n + 9 > 8n + 8 \\ \implies & (n+3)^2 > 8n + 8 \\ \implies & n + 3 > \sqrt{8n+8} \\ \implies & n + 3 - 2\sqrt{2}\sqrt{n+1} > 0. \end{aligned}$$

Therefore the only positive root of $f(k)$ is $\dfrac{n-1}{n}$. □

A.8 Proof of Theorem 3.10

We derive the equilibrium of FPSB$n{\downarrow_p}$ by showing that the game is equivalent to a p-player FPSB game with a particular non-uniform type distribution. Let $q \equiv n/p$. By the definitions of FPSBn (Definition 3.3) and of a reduced game (Definition 3.9), FPSB$n{\downarrow_p}$ is a p-player game with each player having multidimensional type $[v_1, \ldots, v_q]$, with $v_i \sim U[0,1]$ being the type (valuation) of one of the original n players. The key insight for the proof is that since each of the p players in the reduced game must play the same strategy for each of the q players it controls and since the strategy is monotone increasing in type (specifically $k \cdot v_i$), then all but the greatest of the v_i are irrelevant. This is because if one of the q subplayers gets positive payoff it is because it had the highest valuation of $[v_1, \ldots, v_q]$. Thus, each player's type reduces to the maximum of q random variables, i.i.d. $U[0,1]$. In general the cdf of the maximum of m random variables, i.i.d. with cdf F is $F_{\text{MAX}}(x) = F(x)^m$. The cdf for a player's type in FPSB$n{\downarrow_p}$ is the special case, $F_{\text{MAX}}(x) = x^q$.

Theorem 3.1 [McAfee and McMillan, 1987] gives the unique symmetric equilibrium for FPSB with n players having types i.i.d. with cdf F, and a lowest possible type A:

$$a(t) = t - \frac{\int_A^t F(x)^{n-1}\,dx}{F(t)^{n-1}}.$$

Substituting F_{MAX} for F, this gives us the equilibrium for any FPSB (with arbitrary i.i.d. type distribution, F, with lower bound A) reduced from n to p players:

$$a(t) = t - \frac{\int_A^t F(x)^{n-q}\,dx}{F(t)^{n-q}}.$$

In the special case of $U[A,B]$ types, the equilibrium is

$$a(t) = \frac{Ap + n(p-1)}{p + n(p-1)} \cdot t.$$

Finally, for $U[0,1]$, that is, $\text{FPSB}n{\downarrow}_p$, the unique symmetric Nash equilibrium is

$$a(t) = \frac{n(p-1)}{p + n(p-1)} \cdot t. \quad \square$$

A.9 Proof of Lemma 3.11

Corollary 3.6 defines the function $\varepsilon(k) \equiv \varepsilon_{\text{FPSB}n}(k)$. Since we are only concerned with the case $0 < k \leq \frac{n-1}{n}$ and since, by Lemma 3.7, the second and third cases of $\varepsilon(k)$ are equal at $k = \frac{n-1}{n}$ (namely zero) we have for $0 < k \leq \frac{n-1}{n}$ that $\varepsilon(k)$ is equal to the continuously differentiable function f in Lemma 3.7.

To establish that $f(k)$ is strictly decreasing in $(0, \frac{n-1}{n}]$ we first consider the roots of its derivative, $f'(k)$. It is (computationally) easy to differentiate f but takes pages to display. As in the proof of Lemma 3.7 (Appendix A.7) we perform transformations (A.13) on $f'(k)$ that combine common subexpressions and make the coefficients integers. After applying (A.13), simplifying (using the fact that n, a, and b are all positive), and collecting the terms over a common denominator, we can write the numerator of $f'(k)$ as

$$\begin{aligned}
&\left(n^2-1\right)^6 k^{11} + 3\left(n^2-1\right)^5 \left(b^2 + 2a(a+3n+3)\right) k^9 \\
&\quad + 18b^2\left(a(a-3b+3) + 3b + 3(a+b)n\right)\left(n^2-1\right)^4 k^7 - 162ab^4\left(n^2-1\right)^3 k^6 \\
&\quad + 27b^3\left(n^2-1\right)^3 \Big(12(n+1)a^2 + 18\left(b^2 + 2(n+1)^2\right)a + b^2(6n-b+6)\Big)k^5 \\
&\quad - 81b^5\left(n^2-1\right)^2 \Big(b^3 + 2a^2(b-6n-6)) + 12a\left(3b^2 + 2(n+1)b - 3(n+1)^2\right)\Big)k^3 \\
&\quad + 1458ab^6\left(n^2-1\right)^2 k^4 - 4374ab^8\left(n^2-1\right)k^2 \\
&\quad - 486ab^8\left(n^2-1\right)(a + 12(n-b+1))k + 4374ab^{10},
\end{aligned} \tag{A.21}$$

with a and b defined by (A.14).

We can now find the roots of (A.21) given (A.14) by finding the common roots of (A.21), (A.15), and (A.16). As in Appendix A.7 we do this by computing a Gröbner basis, yielding

$$(n-1)^7(n+1)^8k^{13}(kn-n+1)\left(kn^2+n^2-kn+2n+1\right)\left(nk^2-k^2+27n+27\right) \tag{A.22}$$

which, for $n > 1$, has roots at zero,

$$-\frac{n^2+2n+1}{n^2-n}, \qquad \text{and} \tag{A.23}$$

$$\frac{n-1}{n}.$$

These comprise all the possible real roots of the numerator and so there can be no other roots of $f'(k)$. Since (A.23) is negative and the only other possible roots are on the boundaries of $(0, \frac{n-1}{n}]$

we have established that $f'(k)$ is nonzero in the interior of that range. Since it is also continuous it must then be exclusively negative or positive. This implies that the original function, $\varepsilon(k)$, is either strictly increasing or strictly decreasing in $(0, \frac{n-1}{n}]$. Since, by Lemma 3.7, $f(\frac{n-1}{n}) = \varepsilon(\frac{n-1}{n}) = 0$ we can show that $\varepsilon(k)$ is not strictly increasing by showing for some $k \in (0, \frac{n-1}{n})$ that $f(k) > 0$. It is straightforward to compute

$$\lim_{k \to 0} f(k) = \frac{n-1}{2(n+1)}$$

(which is $\geq 1/6$ for all $n > 1$). This establishes that $f(k) = \varepsilon(k)$ decreases somewhere in $(0, \frac{n-1}{n}]$. And since it never changes direction—being continuous and its derivative having no roots in $(0, \frac{n-1}{n})$—it must be strictly decreasing. □

A.10 Proof of Theorem 3.12

Theorem 3.10 and Theorem 3.8 give us the unique symmetric equilibria of $\text{FPSB}n{\downarrow_p}$ and $\text{FPSB}n$ so it suffices to show that

$$\frac{p-1}{p} < \frac{n(p-1)}{p+n(p-1)} < \frac{n-1}{n}.$$

Consider first the left inequality:

$$\begin{aligned}
& p < n \\
\implies & p - n < 0 \\
\implies & p + np - n < np \\
\implies & p + n(p-1) < np \\
\implies & 1 < \frac{np}{p+n(p-1)} && \text{since } n, p \geq 1 \implies p + n(p-1) > 0 \\
\implies & p - 1 < \frac{np(p-1)}{p+n(p-1)} \\
\implies & \frac{p-1}{p} < \frac{n(p-1)}{p+n(p-1)}.
\end{aligned}$$

Similarly for the second inequality,

$$\begin{aligned}
& p < n \\
\implies & 0 < n - p \\
\implies & pn^2 - n^2 < n - p + pn^2 - n^2 \\
\implies & n^2(p-1) < (n-1)(p+n(p-1)) \\
\implies & \frac{n(p-1)}{p+n(p-1)} < \frac{n-1}{n} && \text{since } n \text{ and } p + n(p-1) \text{ are both } > 0. \quad \square
\end{aligned}$$

A.11 Proof of Theorem 3.13

Theorem 3.12 and Lemma 3.11 establish all of Theorem 3.13 except that

$$\frac{n(q-1)}{q+n(q-1)} < \frac{n(p-1)}{p+n(p-1)}$$

which follows directly from $q < p$:

$$
\begin{aligned}
& q < p \\
\implies & -p < -q \\
\implies & -p + pq + npq - np - nq + n < -q + pq + npq - np - nq + n \\
\implies & (q-1)(p+n(p-1)) < (p-1)(q+n(q-1)) \\
\implies & \frac{n(q-1)}{q+n(q-1)} < \frac{n(p-1)}{p+n(p-1)} \quad \text{(since } p+n(p-1) \text{ and } q+n(q-1) \text{ are } > 0\text{).} \quad \square
\end{aligned}
$$

Appendix B

Monte Carlo Best-Response Estimation

Here we describe the simulation technique we use to sanity-check equilibria found by our analytic best-response solver. The approach takes as input an arbitrary payoff function from types and actions of all agents to real-valued payoffs. It also takes an arbitrary strategy function, which is a one-dimensional function from type to action. The strategies and payoffs are represented as arbitrary Mathematica functions. Additionally, it takes an arbitrary probability distribution from which types are drawn. From these three inputs, it computes an empirical best-response function. Figure 2.2 shows an example of our Monte Carlo method finding the best response to a particular strategy (see Section 2.4).

At the core of the method for empirically generating best-response strategies is a simulator that takes the given payoff function, other agent strategies, other agent type distributions, and a particular own type and action. The simulator then repeatedly samples from the other agents' type distributions, computing for each sampled type the actions according to the known other agent strategies. The resulting payoff is then computed for the given own type and action by evaluating the payoff function. Sample statistics for these payoffs are then recorded and the process is repeated for every combination of own type and action, down to a specified granularity. However, certain shortcuts are taken to avoid needless simulation. First, when simulating different actions for a given type, a confidence bound is continually computed using the sampling statistics for the various actions sampled so far. Further simulation is limited to the range of actions within the confidence bound (designated in Figure B.1 by the two larger dots at $a = 0.25$ and $a = 1$). The confidence bounds are determined by performing mean difference tests on pairs of expected payoff sample statistics and considering any actions that fail at the 95% level to be within the confidence bound for best action. It is these confidence bounds that the error bars represent in Figure 2.2. The amount of simulation for each type-action pair is dynamically determined based on the confidence intervals for the expected payoffs. And within the confidence bound of possible actions, the confidence intervals are compared to prioritize the computation for different actions.

Figure B.1 shows the Monte Carlo best-response method sampling possible actions for a specific type. It dynamically limits the actions that it samples from by comparing expected payoffs and finding an interval likely to contain the best action.

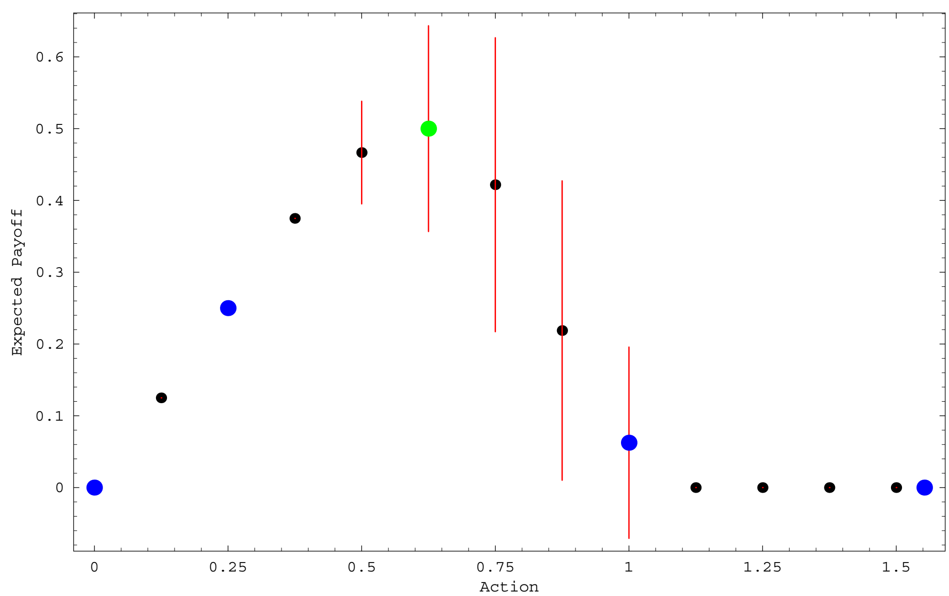

Figure B.1: Empirically estimating the best action for a given type (0) against a given strategy (see Figure 2.2). After < 20 simulations per action the best action is determined, based on the 95% confidence intervals, to be between 0.25 and 1. The maximum likelihood best action is 0.62 (after 100 total samples) and the actual best action is 0.582.

Appendix C

Notes on Equilibria in Symmetric Games

In a symmetric game, every player is identical with respect to the game rules (Definition 1.2). We include here three theorems about the existence of equilibria in symmetric games. (1) A symmetric 2-strategy game must have a pure-strategy Nash equilibrium. (2) As a special case of a theorem in Nash's [1951] seminal paper, any finite symmetric game has a symmetric Nash equilibrium.[1] (3) Symmetric infinite games with compact, convex strategy spaces and continuous, quasiconcave utility functions have symmetric pure-strategy Nash equilibria.

Theorem C.1 *A symmetric game with two strategies has an equilibrium in pure strategies.*

Proof. Let $S = \{1, 2\}$ and let $[i, n-i]$ denote the profile with $i \in \{0, \ldots, n\}$ players playing strategy 1 and $n - i$ playing strategy 2. Let $u_s^i \equiv u(s, [i, n-i])$ be the payoff to a player playing strategy $s \in S$ in the profile $[i, n-i]$. Define the boolean function $pe(i)$ as follows:

$$pe(i) \equiv \begin{cases} u_2^0 \geq u_1^1 & \text{if } i = 0 \\ u_1^n \geq u_2^{n-1} & \text{if } i = n \\ u_1^i \geq u_2^{i-1} \text{ \& } u_2^i \geq u_1^{i+1} & \text{if } i \in \{1 \ldots n-1\}. \end{cases}$$

In words, $pe(i) = \text{TRUE}$ when no unilateral deviation from $[i, n-i]$ is beneficial—i.e., when $[i, n-i]$ is a pure-strategy equilibrium.

Assuming the opposite of what we want to prove, $pe(i) = \text{FALSE}$ for all $i \in \{0, \ldots, n\}$. We first show by induction on i that $u_2^i < u_1^{i+1}$ for all $i \in \{0, \ldots, n-1\}$. The base case, $i = 0$, follows directly from $pe(0) = \text{FALSE}$. For the general case, suppose $u_2^k < u_1^{k+1}$ for some $k \in \{0, \ldots, n-2\}$. Since $pe(k+1) = \text{FALSE}$, we have $u_1^{k+1} < u_2^k$ or $u_2^{k+1} < u_1^{k+2}$. The first disjunct contradicts the inductive hypothesis, implying $u_2^{k+1} < u_1^{k+2}$, which concludes the inductive proof. In particular, $u_2^{n-1} < u_1^n$. But, $pe(n) = \text{FALSE}$ implies $u_1^n < u_2^{n-1}$, and we have a contradiction. Therefore $pe(i) = \text{TRUE}$ for some i. □

Can we relax the sufficient conditions of Theorem C.1? First, consider symmetric games with more than two strategies. Rock-Paper-Scissors (Table C.1a) is a counterexample showing that the theorem no longer holds. It is a three-strategy symmetric game with no pure-strategy equilibrium. This is the case because, of the six pure-strategy profiles (RR, RP, RS, PP, PS, SS), none constitute an equilibrium.

[1] We also discuss here Nash's generalized notion of symmetry in games.

Table C.1: (a) Rock-Paper-Scissors: a 3-strategy symmetric game with no pure-strategy equilibria. (b) Matching Pennies: a 2-strategy asymmetric game with no pure-strategy equilibria. (c) Anti-coordination Game: a symmetric game with no symmetric pure equilibria.

	R	**P**	**S**
R	0,0	-1,1	1,-1
P	1,-1	0,0	-1,1
S	-1,1	1,-1	0,0

(a)

	H	**T**
H	1,0	0,1
T	0,1	1,0

(b)

	A	**B**
A	0,0	1,1
B	1,1	0,0

(c)

Next, does the theorem apply to *asymmetric* two-strategy games? Again, no, as demonstrated by Matching Pennies (Table C.1b). In Matching Pennies, both players simultaneously choose an action—heads or tails—and player 1 wins if the actions match and player 2 wins otherwise. Again, none of the pure-strategy profiles (HH,TT,HT,TH) constitute an equilibrium.

Finally, can we strengthen the conclusion of Theorem C.1 to guarantee *symmetric* pure-strategy equilibria? A simple anti-coordination game (Table C.1c) serves as a counterexample. In this game, each player receives 1 when the players choose different actions and 0 otherwise. The only pure-strategy equilibria are the profiles where the players choose different actions, i.e., asymmetric profiles.

We next consider other sufficient conditions for pure and/or symmetric equilibria in symmetric games. First we present a lemma establishing properties of the mapping from profiles to best responses. The result is analogous to Lemma 8.AA.1 in Mas-Colell *et al.* [1995], adapted to the case of symmetric games.

Lemma C.2 *In a symmetric game $[n, S, u()]$ with S nonempty, compact, and convex and with $u(s_i, s_1, \ldots, s_n)$ continuous in $(s_1, \ldots, s_n)$ and quasi-concave in s_i, the best response correspondence*

$$\begin{aligned} b(s) &\equiv \arg\max_{t \in S} u(t, s) \\ &= \{\tau : u(\tau, s) = \max_{t \in S} u(t, s)\} \end{aligned}$$

is nonempty, convex-valued, and upper hemicontinuous.

Proof. Since $u(S, s)$ is the continuous image of the compact set S, it is compact and has a maximum and so $b(s)$ is nonempty. $b(S)$ is convex because the set of maxima of a quasiconcave function ($u(\cdot, s)$) on a convex set (S) is convex. To show that $b(\cdot)$ is upper hemicontinuous, show that for any sequence $s^n \to s$ such that $s^n \in b(s^n)$ for all n, we have $s \in b(s)$. To see this, note that for all n, $u(s_i^n, s^n) \geq u(s_i', s^n)$ for all $s_i' \in S$. So by continuity of $u(\cdot)$, we have $u(s_i, s) \geq u(s_i', s)$. □

We now show our second main result, that infinite symmetric games with certain properties have symmetric equilibria in pure strategies. This result corresponds to Proposition 8.D.3 in Mas-Colell *et al.* [1995] which establishes the existence of (possibly asymmetric) pure strategy equilibria for the corresponding class of possibly asymmetric games.

Theorem C.3 *A symmetric game* $[n, S, u()]$ *with* S *a nonempty, convex, and compact subset of some Euclidean space and* $u(s_i, s_1, \ldots, s_n)$ *continuous in* $(s_1, \ldots, s_n)$ *and quasiconcave in* s_i *has a symmetric pure-strategy equilibrium.*

Proof. $b(\cdot)$ is a correspondence from the nonempty, convex, compact set S to itself. By Lemma C.2, $b(\cdot)$ is a nonempty, convex-valued, and upper hemicontinuous correspondence. Thus, by Kakutani's Theorem[2], there exists a fixed point of $b(\cdot)$ which implies that there exists an $s \in S$ such that $s \in b(s)$ and so all playing s is a symmetric equilibrium. □

Finally, we present Nash's [1951] result that finite symmetric games have symmetric equilibria.[3] This is a special case of his result that every finite game has a "symmetric" equilibrium, where Nash's definition of a symmetric profile is one invariant under every automorphism of the game. This turns out to be equivalent to defining a symmetric profile as one in which all the symmetric players (if any) are playing the same mixed strategy. In the case of a symmetric game, the two notions of a symmetric profile (invariant under automorphisms vs. simply homogeneous) coincide and we have our result.

Here we present an alternate proof of the result, modeled on Nash's seminal proof of the existence of (possibly asymmetric) equilibria for general finite games.[4]

Theorem C.4 *A finite symmetric game has a symmetric mixed-strategy equilibrium.*

Proof. For each pure strategy $s \in S$, define a continuous function of a mixed strategy σ by

$$g_s(\sigma) \equiv \max(0, u(s, \sigma) - u(\sigma, \sigma)).$$

In words, $g_s(\sigma)$ is the gain, if any, of unilaterally deviating from the symmetric mixed profile of all playing σ to playing pure strategy s. Next define

$$y_s(\sigma) \equiv \frac{\sigma_s + g_s(\sigma)}{1 + \sum_{t \in S} g_t(\sigma)}.$$

The set of functions $y_s(\cdot) \forall s \in S$ defines a mapping from the set of mixed strategies to itself. We first show that the fixed points of $y(\cdot)$ are equilibria. Of all the pure strategies in the support of σ, one, say w, must be worst, implying $u(w, \sigma) \leq u(\sigma, \sigma)$ which implies that $g_w(\sigma) = 0$.

Assume $y(\sigma) = \sigma$. Then y must not decrease σ_w. The numerator is σ_w so the denominator must be 1, which implies that for all $s \in S$, $g_s(\sigma) = 0$ and so all playing σ is an equilibrium.

Conversely, if all playing σ is an equilibrium then all the g's vanish, making σ a fixed point under $y(\cdot)$.

Finally, since $y(\cdot)$ is a continuous mapping of a compact, convex set, it has a fixed point by Brouwer's Theorem. □

---

[2] For this and other fixed point theorems (namely Brouwer's, used in Theorem C.4) see, for example, the mathematical appendix (p952) of Mas-Colell *et al.* [1995].

[3] This result can also be proved as a corollary to Theorem C.3 by viewing a finite game as an infinite game with strategy sets being possible mixtures of pure strategies.

[4] In Nash's [1950] PhD thesis, he proved this theorem using his contemporary Kakutani's generalization of Brouwer's fixed-point theorem to the case of correspondences (point-to-set mappings) but in a subsequent publication [Nash, 1951] he presented a simplified proof that required only Brouwer's fixed-point theorem. Theorem C.4 is modeled on the simplified proof.

Since exploitation of symmetry has been known to produce dramatic simplifications of optimization problems [Boyd, 1990] it is somewhat surprising that it has not been actively addressed in the game theory literature. In addition to the methods we have used for game solving and that we discuss in Section 3.6 (replicator dynamics and function minimization with Amoeba), we believe symmetry may be productively exploited for other algorithms as well. For example, the well-known Lemke-Howson algorithm for 2-player (bimatrix) games lends itself to a simplification if the game is symmetric and if we restrict our attention to symmetric equilibria: we need only one payoff matrix, one mixed strategy vector, and one slackness vector, reducing the size of the Linear Complementarity Problem by a factor of two. As we discuss in Section 3.6, exploiting symmetry in games with more than two players can yield enormous computational savings. Finally, as noted in Section 3.9, Bhat and Leyton-Brown [2004] have recently published an algorithm for solving a class of games which they extend to exploit symmetry.

Appendix D

Strategies for SAA Experiments

This appendix describes in detail the strategies used in our experiments in Chapter 5, in particular the price prediction strategies. Table D.1 lists all the strategies we consider.

D.1 Price Predicting Strategies

Agents using price prediction strategies generate an initial, pre-auction belief about the final prices of the goods. For example, the agent can believe that the final prices will all equal zero or they are distributed normally or uniformly on some interval. The derivation of most beliefs we employ involves Monte Carlo sampling. To obtain a belief for a particular distribution of agents' preferences, we simulate a large number of game instances with agents drawn from the preference distribution.

In most of our reported results we use the uniform and exponential distributions of number of goods demanded by the agent. We have also considered agents who demanded a fixed number of goods. We refer to such a preference distribution as a fixed distribution. (See Section 5.1 for details on preference (type) distributions in SAA.)

We consider two families of price predictors (Section 4.5). The *point prediction* strategy family has point beliefs about the final prices that will be realized for each good. The point price predicting strategy is thus parameterized by its vector of point beliefs. We label such beliefs $\boldsymbol{\pi}(\varnothing)$. Any price vector can represent initial beliefs. Point beliefs based on sampling are obtained by averaging across final prices or demands in the simulated games.

The point prediction family includes a sub-class of strategies with participation-only prediction. The idea behind this strategy is that it ignores its predictions at some stages of decision making (unlike the full point predictor, which always relies on the predicted vector). We describe this strategy in Section 5.5.

The *distribution prediction* strategy family has beliefs about the final price distributions. Let $F \equiv F(\varnothing)$ denote a joint cumulative distribution function over final prices, representing the agent's pre-auction belief. We assume that prices are bounded above by a known constant, V. Thus, F associates cumulative probabilities with price vectors in $\{0, \ldots, V\}^m$. For simplicity we use only the information contained in the vector of marginal distributions, $(F_1, \ldots, F_m)$, as if the final prices are independent across goods.

For technical reasons we require that initial beliefs are such that for each good all prices in $\{0, \ldots, V\}$ have a positive probability of occurring. To ensure that this requirement is satisfied for initial beliefs obtained from empirical samples, we modify the latter before constructing beliefs. In

Strategy Family and Notation	Family Parameter	Examples
Straightforward bidder, SB	N/A	SB
Sunk-Aware agent, SA(k)	Sunk-awareness parameter, k	SA(k) $k = 0, 0.05, 0.1, \dots 0.95$
Point Price Predictors, $\mathrm{PP}(\boldsymbol{\pi}^x)$	Beliefs about average final prices of the goods	$\mathrm{PP}(\boldsymbol{\pi}^{Zero})$ $\mathrm{PP}(\boldsymbol{\pi}^{\infty})$ $\mathrm{PP}(\boldsymbol{\pi}^{ECE})$ $\mathrm{PP}(\boldsymbol{\pi}^{EDCE})$ $\mathrm{PP}(\boldsymbol{\pi}^{SB})$ $\mathrm{PP}(\boldsymbol{\pi}^{SC})$
Point Price Predictors with participation only, $\mathrm{PP}_{\mathrm{po}}(\boldsymbol{\pi}^x)$	Beliefs about average final prices of the goods	$\mathrm{PP}_{\mathrm{po}}(\boldsymbol{\pi}^{\infty})$ $\mathrm{PP}_{\mathrm{po}}(\boldsymbol{\pi}^{Zero})$ $\mathrm{PP}_{\mathrm{po}}(\boldsymbol{\pi}^{ECE})$ $\mathrm{PP}_{\mathrm{po}}(\boldsymbol{\pi}^{EDCE})$ $\mathrm{PP}_{\mathrm{po}}(\boldsymbol{\pi}^{SB})$ $\mathrm{PP}_{\mathrm{po}}(\boldsymbol{\pi}^{SC})$
Price Distribution Predictors, $\mathrm{PP}(F^x)$ $\mathrm{PP}(G(\mu(x), \sigma(y)))$ $\mathrm{PP}(F(\boldsymbol{\pi}^x))$	Beliefs about final price distributions; such beliefs are labeled by x or a pair $(\boldsymbol{\mu}, \boldsymbol{\sigma})$	$\mathrm{PP}(F^{Zero})$ $\mathrm{PP}(F^{U})$ $\mathrm{PP}(F^{SB})$ $\mathrm{PP}(F^{CE})$ $\mathrm{PP}(G(\mu(CE), \sigma(CE)))$ $\mathrm{PP}(G(\mu(SB), \sigma(SB)))$ $\mathrm{PP}(F(\boldsymbol{\pi}^{ECE}))$ $\mathrm{PP}(F(\boldsymbol{\pi}^{EDCE}))$ $\mathrm{PP}(F(\boldsymbol{\pi}^{SB}))$ $\mathrm{PP}(F(\boldsymbol{\pi}^{SC}))$

Table D.1: Reference table. The strategy families we have studied are listed in column 1 and their parameters are in column 2. Column 3 lists strategies we have employed in many of our restricted games.

particular, we add to the empirically obtained sample of prices a sample of (V + 1) prices from 0 to V. Suppose the empirical sample consists of N games. The probability that the good will have final price $p \in \{0, \ldots, V\}$ is then

$$\Pr(p) = \frac{N_p + 1}{N + (V + 1)}, \tag{D.1}$$

where N_p is the number of times the final price equaled p in the original sample. If N is high relative to V, the effect of adding an artificial sample to the empirical one is negligible. In the restricted games we studied $V = 50$ and $N \geq 340{,}000$.

In the following sections we describe some examples of beliefs. We denote a specific point price prediction strategy by $\mathrm{PP}(\boldsymbol{\pi}^x)$, where x labels particular initial point beliefs, $\boldsymbol{\pi}(\varnothing)$. We denote the strategy of bidding based on a particular distribution predictor by $\mathrm{PP}(F^x)$, where x labels various initial beliefs about final price distributions, $F(\varnothing)$. If the initial beliefs x are based on Monte Carlo sampling, we write x_u and x_e to distinguish between beliefs obtained using draws from uniform and exponential preference distributions.[1]

Zero Beliefs

To construct zero beliefs for the distribution predictor, we assume that we have a sample of $n = 10^6$ zero final prices. After we add another artificial sample of $(V + 1) = 51$ prices (see Equation D.1), the probability that the final price of a good will be zero becomes $1{,}000{,}000/1{,}000{,}051 = 0.99995$. The probability that the final price of this good will equal $p \in [1, V]$ becomes $1/1{,}000{,}051 < 10^{-6}$. Note, however, that as soon as the ask price exceeds zero, the distribution predictor reconditions its beliefs based on that information. So once all the ask prices are above zero it bids identically to a distribution predictor having a uniform distribution for final prices.

For the point price predictor, zero beliefs are simply a vector of zeros. The point price predictor with zero beliefs is equivalent to SB. We denote the distribution predictors using zero beliefs by $\mathrm{PP}(F^{Zero})$. A point predictor with zero beliefs is $\mathrm{PP}(\mathbf{0})$, or, for consistency, $\mathrm{PP}(\boldsymbol{\pi}^{Zero})$, or of course simply SB.

Infinite Point Beliefs

Infinite point beliefs are defined only for the point price predictor. We label such a predictor $\mathrm{PP}(\boldsymbol{\pi}^{\infty})$. We define this strategy so that it reverts to SB if the agent has single-unit preference. Since there is no exposure problem, such an agent will bid if and only if it has a positive value for exactly one of the goods (despite the expectation that the price will be infinite). Thus, this strategy represents the extreme of conservative bidding, never risking exposure. The performance of this strategy provides a useful performance benchmark for price predicting strategies.

Uniform Distribution Beliefs

The uniform distribution beliefs are defined only for the distribution predictor, which we denote $\mathrm{PP}(F^U)$. According to the uniform beliefs, the probability that the final price of a good will equal $p \in [0, V]$ is $1/51 = 0.0196$.

---

[1] We suppress this subscript when it is clear from context (see Section D.3).

Baseline Beliefs

Baseline beliefs are based on the final prices in a sample of games in which all players use the SB strategy. The baseline point beliefs are a vector of average final prices of the goods available. The baseline distribution beliefs are a vector of marginal price distributions computed according to Equation D.1. Our sample size is $N = 10^6$ games, and the upper bound on prices is $V = 50$ for all goods.

We denote the point predictors with SB beliefs derived using samples from the uniform and exponential distributions by $\mathrm{PP}(\boldsymbol{\pi}^{SB_u})$ and $\mathrm{PP}(\boldsymbol{\pi}^{SB_e})$ respectively. Similarly, the distribution predictors with SB beliefs are labeled $\mathrm{PP}(F^{SB_u})$ and $\mathrm{PP}(F^{SB_e})$.

Walrasian Equilibrium Beliefs

Suppose that the final prices form a Walrasian price equilibrium (PE) in the SAA game. This is guaranteed, for example, when SB agents all demand only single goods, but is not true in general. However, in our experience, the final prices are generally not too far from PE. Therefore, we calculate the Walrasian equilibrium for an SAA environment and use the resulting prices to create initial beliefs.

Let SE be an SAA environment. To find the vector of *Walrasian equilibrium price distributions* for SE, F^{PE}, we randomly generate many ($N = 25000$) game instances with agents drawn from the preference distribution of SE, and use tatonnement to solve for the equilibrium prices in each. (See Chapter 4 for details.) The PE distribution beliefs are computed based on the sample of the equilibrium prices according to Equation D.1. We label the PE distribution predictor as $\mathrm{PP}(F^{PE_u})$ if the preference distribution of the sample is uniform, and as $\mathrm{PP}(F^{PE_e})$ if it is exponential.

We calculate the *expected price equilibrium* beliefs, $\boldsymbol{\pi}^{EPE}$, by averaging across the prices in the sample. We calculate the *expected demand price equilibrium* beliefs, $\boldsymbol{\pi}^{EDPE}$, by calculating the expected demand function for each of the game instances, and then solving for the Walrasian equilibrium based on the average demands. We label the corresponding bid strategies as $\mathrm{PP}(\boldsymbol{\pi}^{EPE_u})$ and $\mathrm{PP}(\boldsymbol{\pi}^{EDPE_u})$ if the preference distribution of the sample is uniform, and as $\mathrm{PP}(\boldsymbol{\pi}^{EPE_e})$ and $\mathrm{PP}(\boldsymbol{\pi}^{EDPE_e})$ if it is exponential.[2] We describe the Walrasian prediction approach in greater detail in Chapter 4.

Self-Confirming Prices

The self-confirming (SC) distribution beliefs are what we call self-confirming marginal price distributions (Section 4.6). Let SE be an SAA environment. The prediction $F = (F_1, \ldots, F_m)$ is a vector of *self-confirming marginal price distributions for* SE iff for all i, F_i is the marginal distribution of prices for good i resulting when all agents play bidding strategy $\mathrm{PP}(F)$ in SE.

The SC point beliefs are what we call *self-confirming point predictions*, which are defined as a vector of point predictions ($\boldsymbol{\pi}$) that on average are correct, if all agents use point price prediction. Note that the mean of an SC distribution may be different from SC point predictions for the same environment.

The general idea behind the derivation algorithms of both types of self-confirming predictions is as follows. Given an SAA environment, we derive self-confirming predictions through an iterative

[2]The prices to which tatonnement converges are sensitive to the choice of initial prices and other parameters of the algorithm. In Section D.2 we list all the price vectors we obtained for the 5×5 uniform environment. We have not derived PE beliefs for alternative environments.

simulation process. Starting from an arbitrary prediction, we run many instances (N) of an SAA environment (sampling from the given preference distributions) with all agents playing the same predicting strategy (either point prediction or distribution prediction). We record the resulting prices from each instance, and create new beliefs for the predictors. The new point beliefs are a vector of average final prices from the first iteration. The new distribution beliefs are the sample distribution. We repeat the process using the new beliefs for each new iteration. If it ever reaches an approximate fixed point, then we have statistically identified approximate self-confirming predictions for this environment. We describe further details of SC point prediction in Section 4.6 and SC distribution prediction there and in other work [Osepayshvili *et al.*, 2005].

For the environments we investigate, we can find both SC point predictions and distributions. In our experiments, $N = 500{,}000$ for SC point predictions and $N = 10^6$ for SC distributions; the initial beliefs used in the first iteration are zero beliefs (Section D.1), although our results do not appear sensitive to this.

Gaussian Distribution Beliefs

The Gaussian distribution beliefs are defined only for the distribution predictor. Suppose we know the expected prices $\boldsymbol{\mu}$ and the standard deviations $\boldsymbol{\sigma}$ for all the goods. Then we can approximate the final price distribution of good i with a Gaussian centered on μ_i with the restriction that prices $p \in [0, V]$. To implement Gaussian beliefs we draw a random sample of size $N = 1000000$ from $N(\mu_i, \sigma_i)$ for each good i and discard prices outside $[0, V]$.[3] Then we compute the final price probabilities according to Equation D.1. We denote distribution predictors with such beliefs by $G(\boldsymbol{\mu}, \boldsymbol{\sigma})$, where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ label the expected prices and standard deviations. For example, strategy $\mathrm{PP}(G(\mu(PE_u), \sigma(PE_u)))$ has Gaussian beliefs created using the means and standard deviations of the distribution vector F^{PE_u}; agent $\mathrm{PP}(G(\boldsymbol{\pi}^{EDPE_u}, \sigma(PE_u)))$ has Gaussian beliefs created using $\boldsymbol{\pi}^{EDPE_u}$ as the means vector and the standard deviations of the distribution vector F^{PE_u}.

Degenerate Distribution Beliefs

The degenerate distribution beliefs are defined only for the distribution predictor. They are degenerate in the sense that according to such beliefs each good has a deterministic final price. Let $\boldsymbol{\pi}$ be a vector of prices. For every good i, assign probability 1 to the ith element of $\boldsymbol{\pi}$ and probability 0 to all other prices in $[0, V]$. We denote the distribution predictor with such beliefs by $\mathrm{PP}(F(\boldsymbol{\pi}))$. If we round the vector $\boldsymbol{\pi}$ before generating the distributions, we mark the vector by a prime in the strategy name: $\mathrm{PP}(F(\boldsymbol{\pi}'))$. For example, $\mathrm{PP}(F(\boldsymbol{\pi}^{SB_u}))$ has beliefs that the final prices will equal $\boldsymbol{\pi}^{SB_u}$ with probability 1; $\mathrm{PP}(F(\boldsymbol{\pi}^{EPE'_u}))$ has beliefs that the final prices will equal the rounded elements of $\boldsymbol{\pi}^{EPE_u}$ with probability 1. The former beliefs match better the information that the corresponding point predictor has. The latter beliefs are justified by the fact that prices are integers in our model.

D.2 53-Strategy Game For 5×5 Uniform Environment

We constructed 53 strategies for the 5×5 uniform environment (Section 5.7). In Table D.2 we list all the strategies, relevant sections of this appendix and all our publications in which the strategies

[3] One drawback of this approach is that truncating the tails shifts the means of the distribution.

Strategy Family	#	Strategy Notation
N/A	1	SB
Sunk-Aware	20	SA(k) $k = 0, 0.05, 0.1, \ldots 0.95$
Point Price Predictors	13	PP($\boldsymbol{\pi}^{\infty}$), PP($\boldsymbol{\pi}^{SB_u}$), PP$_{\mathrm{po}}$($\boldsymbol{\pi}^{SB_u}$), PP($\boldsymbol{\pi}^{EPE_u}$), PP($\boldsymbol{\pi}^{EPE^*_u}$), PP($\boldsymbol{\pi}^{EPE^{**}_u}$), PP($\boldsymbol{\pi}^{EPE_e}$), PP($\boldsymbol{\pi}^{EPE^*_e}$), PP($\boldsymbol{\pi}^{EDPE_u}$), PP($\boldsymbol{\pi}^{EDPE^*_u}$), PP($\boldsymbol{\pi}^{EDPE_e}$), PP($\boldsymbol{\pi}^{EDPE^*_e}$), PP($\boldsymbol{\pi}^{SC_u}$)
Price Distribution Predictors	19	PP(F^{Zero}),PP(F^U) PP(F^{SB_u}),PP(F^{SC_u}), PP(F^{PE_u}), $\mathrm{PP}(G(\mu(PE_u), \sigma(PE_u)))$, $\mathrm{PP}(G(\mu(SB_u), \sigma(SB_u)))$, $\mathrm{PP}(G(\boldsymbol{\pi}^{EDPE_u}, \sigma(PE_u)))$, $\mathrm{PP}(G(\boldsymbol{\pi}^{EDPE_u}, \sigma(SB_u)))$, $\mathrm{PP}(G(\boldsymbol{\pi}^{SC_u}, \sigma(PE_u)))$, $\mathrm{PP}(G(\boldsymbol{\pi}^{SC_u}, \sigma(SB_u)))$, $\mathrm{PP}(F(\boldsymbol{\pi}^{EPE_u}))$, $\mathrm{PP}(F(\boldsymbol{\pi}^{EPE'_u}))$, $\mathrm{PP}(F(\boldsymbol{\pi}^{EDPE_u}))$, $\mathrm{PP}(F(\boldsymbol{\pi}^{EDPE'_u}))$, $\mathrm{PP}(F(\boldsymbol{\pi}^{SB_u}))$, $\mathrm{PP}(F(\boldsymbol{\pi}^{SB'_u}))$ $\mathrm{PP}(F(\boldsymbol{\pi}^{SC_u}))$, $\mathrm{PP}(F(\boldsymbol{\pi}^{SC'_u}))$

Table D.2: 53 strategies for the largest 5×5 uniform environment game. The strategies are listed in column 3, and their strategy families are given in column 1. The po subscript refers to participation only prediction. Different point beliefs created by the same (non-deterministic) algorithm are marked by asterisks. Column 2 gives the total number of strategies from a particular family represented in the 53-strategy game.

were mentioned. Table D.3 presents all initial point beliefs for the price predicting strategies we derived for the 5×5 uniform environment, as well as relevant sections of this appendix.

D.3 Strategies For Alternative Environments

Table D.4 describes the pool of 27 strategies that we searched for a profitable deviation from playing $\mathrm{PP}(F^{SC})$ when all the other agents play $\mathrm{PP}(F^{SC})$. Table D.5 describes 7-strategy games (7-cliques) for each of the alternative environments we consider (Section 5.7). Some of the strategies are price predictors whose initial beliefs are derived using Monte Carlo sampling. Such beliefs are therefore parameterized by the underlying preference distribution. However, we suppress the preference distribution labels in the tables, because we derived these beliefs only for the environment in which the corresponding predicting strategy was used, and therefore there is no ambiguity about how the beliefs were derived.

Beliefs/Good	Initial Point Beliefs					Appendix Section
$\boldsymbol{\pi}^{\infty}$	1000	1000	1000	1000	1000	D.1
$\boldsymbol{\pi}^{SB_u}$	14.8	10.7	7.6	4.6	1.9	D.1
$\boldsymbol{\pi}^{SC_u}$	13.0	8.7	5.4	3.0	1.2	D.1
$\boldsymbol{\pi}^{EPE_u}$	16.6	10.8	6.5	3.1	0.7	D.1
$\boldsymbol{\pi}^{EPE_u^*}$	16.5	10.7	6.4	3.1	0.8	D.1
$\boldsymbol{\pi}^{EPE_u^{**}}$	26.0	14.2	6.9	2.5	0.3	D.1
$\boldsymbol{\pi}^{EPE_e}$	6.0	4.1	1.8	0.6	0.1	D.1
$\boldsymbol{\pi}^{EPE_e^*}$	30.5	11.9	6.0	2.7	0.4	D.1
$\boldsymbol{\pi}^{EDPE_u}$	20.0	12.0	8.0	2.0	0.0	D.1
$\boldsymbol{\pi}^{EDPE_u^*}$	20.8	11.4	8.2	1.8	0.0	D.1
$\boldsymbol{\pi}^{EDPE_e}$	25.0	10.0	5.1	0.9	0.0	D.1
$\boldsymbol{\pi}^{EDPE_e^*}$	24.5	10.5	5.5	1.5	0.0	D.1

Table D.3: Initial beliefs (rounded to one decimal place) for the 5×5 uniform environment. The notation for the beliefs is presented in column 1. The vectors of point beliefs for the five goods are presented in column 2. The monotonicity of the prices is due to the specifics of the preference distribution (Section 5.1). The subsections of Section D.1 that are most relevant to a particular belief are presented in column 3.

Strategy Family	#	Strategy Notation
N/A	1	SB
Sunk- Aware	20	SA(k) $k = 0, 0.05, 0.1, \ldots 0.95$
Point Price Predictors	3	PP($\boldsymbol{\pi}^{\infty}$), PP($\boldsymbol{\pi}^{SB}$), PP($\boldsymbol{\pi}^{SC}$)
Price Distribution Predictors	3	PP(F^U), PP(F^{SB}), PP(F^{SC})

Table D.4: Pool of 27 deviators for alternative environments. The strategies are listed in column 3, and their strategy families are given in column 1. Column 2 gives then total number of strategies from a particular family represented in the 27-strategy pool.

Environment	Strategies
$E(3,3)$	$\mathrm{PP}(F^{SC})$, $\mathrm{PP}(F^{SB})$, SA(0.6), SA(0.7), SA(0.75), SA(0.8), SA(0.85)
$E(3,5)$	$\mathrm{PP}(F^{SC})$, $\mathrm{PP}(F^{SB})$, $\mathrm{PP}(F^{U})$, $\mathrm{PP}(\boldsymbol{\pi}^{SC})$, $\mathrm{PP}(\boldsymbol{\pi}^{SB})$, $\mathrm{PP}(\boldsymbol{\pi}^{\infty})$, SA(0.85)
$E(3,8)$	$\mathrm{PP}(F^{SC})$, $\mathrm{PP}(F^{SB})$, $\mathrm{PP}(F^{U})$, $\mathrm{PP}(\boldsymbol{\pi}^{SC})$, $\mathrm{PP}(\boldsymbol{\pi}^{SB})$, $\mathrm{PP}(\boldsymbol{\pi}^{\infty})$, SA(0.85)
$E(5,3)$	$\mathrm{PP}(F^{SC})$, $\mathrm{PP}(F^{SB})$, $\mathrm{PP}(F^{U})$, $\mathrm{PP}(\boldsymbol{\pi}^{SC})$, $\mathrm{PP}(\boldsymbol{\pi}^{SB})$, $\mathrm{PP}(\boldsymbol{\pi}^{\infty})$, SB
$E(5,5)$	$\mathrm{PP}(F^{SC})$, $\mathrm{PP}(F^{SB})$, $\mathrm{PP}(F^{U})$, $\mathrm{PP}(\boldsymbol{\pi}^{SC})$, $\mathrm{PP}(\boldsymbol{\pi}^{SB})$, $\mathrm{PP}(\boldsymbol{\pi}^{\infty})$, SA(0.35)
$E(5,8)$	$\mathrm{PP}(F^{SC})$, $\mathrm{PP}(F^{SB})$, $\mathrm{PP}(F^{U})$, $\mathrm{PP}(\boldsymbol{\pi}^{SC})$, $\mathrm{PP}(\boldsymbol{\pi}^{SB})$, $\mathrm{PP}(\boldsymbol{\pi}^{\infty})$, SA(0.35)
$E(7,3)$	$\mathrm{PP}(F^{SC})$, $\mathrm{PP}(F^{SB})$, SA(0.55), SA(0.65), SA(0.7), SA(0.75), SA(0.8)
$E(7,6)$	$\mathrm{PP}(F^{SC})$, $\mathrm{PP}(F^{SB})$, $\mathrm{PP}(F^{U})$, $\mathrm{PP}(\boldsymbol{\pi}^{SC})$, $\mathrm{PP}(\boldsymbol{\pi}^{SB})$, $\mathrm{PP}(\boldsymbol{\pi}^{\infty})$, SB
$U(3,3)$	$\mathrm{PP}(F^{SC})$, $\mathrm{PP}(F^{SB})$, SA(0.65), SA(0.7), SA(0.75), SA(0.8), SA(0.85)
$U(3,5)$	$\mathrm{PP}(F^{SC})$, $\mathrm{PP}(F^{SB})$, $\mathrm{PP}(\boldsymbol{\pi}^{SC})$, $\mathrm{PP}(\boldsymbol{\pi}^{SB})$, SA(0.7), SA(0.75), SA(0.85)
$U(3,8)$	$\mathrm{PP}(F^{SC})$, $\mathrm{PP}(F^{SB})$, $\mathrm{PP}(F^{U})$, $\mathrm{PP}(\boldsymbol{\pi}^{SC})$, $\mathrm{PP}(\boldsymbol{\pi}^{SB})$, $\mathrm{PP}(\boldsymbol{\pi}^{\infty})$, SA(0.9)
$U(5,3)$	$\mathrm{PP}(F^{SC})$, $\mathrm{PP}(F^{SB})$, SA(0.6), SA(0.65), SA(0.7), SA(0.75), SA(0.8)
$U(5,8)$	$\mathrm{PP}(F^{SC})$, $\mathrm{PP}(F^{SB})$, $\mathrm{PP}(F^{U})$, $\mathrm{PP}(\boldsymbol{\pi}^{SC})$, $\mathrm{PP}(\boldsymbol{\pi}^{SB})$, $\mathrm{PP}(\boldsymbol{\pi}^{\infty})$, SA(0.9)
$U(7,3)$	$\mathrm{PP}(F^{SC})$, $\mathrm{PP}(F^{SB})$, SA(0.55), SA(0.65), SA(0.7), SA(0.75), SA(0.8)
$U(7,6)$	$\mathrm{PP}(F^{SC})$, $\mathrm{PP}(F^{SB})$, $\mathrm{PP}(\boldsymbol{\pi}^{SC})$, $\mathrm{PP}(\boldsymbol{\pi}^{SB})$, SA(0.75), SA(0.8), SA(0.9)
$U(7,8)$	$\mathrm{PP}(F^{SC})$, $\mathrm{PP}(F^{SB})$, $\mathrm{PP}(F^{U})$, $\mathrm{PP}(\boldsymbol{\pi}^{SC})$, $\mathrm{PP}(\boldsymbol{\pi}^{SB})$, $\mathrm{PP}(\boldsymbol{\pi}^{\infty})$, SB

Table D.5: 7-strategy games for alternative environments. $E(m,n)$ and $U(m,n)$ refer to various $\mathrm{SAA}_{\lambda\sim U}$ and $\mathrm{SAA}_{\lambda\sim E}$ environments.

Bibliography

Olivier Armantier, Jean-Pierre Florens, and Jean-Francois Richard. Empirical game-theoretic models: Constrained equilibrium and simulations. Technical report, State University of New York at Stonybrook, 2000.

Olivier Armantier, Jean-Pierre Florens, and Jean-Francois Richard. Approximation of Bayesian Nash equilibria. *Games and Economic Behavior*, to appear.

Kenneth J. Arrow and Frank H. Hahn. *General Competitive Analysis*. Holden-Day, San Francisco, 1971.

W. Brian Arthur, John H. Holland, Blake LeBaron, Richard Palmer, and Paul Tayler. Asset pricing under endogenous expectations in an artificial stock market. In W. Brian Arthur, Steven N. Durlauf, and David A. Lane, editors, *The Economy as an Evolving Complex System II*. Addison-Wesley, 1997.

Susan Athey. Single crossing properties and the existence of pure strategy equilibria in games of incomplete information. *Econometrica*, 69(4):861–890, 2001.

Robert J. Aumann and Sergiu Hart. *Handbook of Game Theory*, volume 1. North-Holland, 1992.

Thomas Becker and Volker Weispfenning. *Gröbner Bases: A Computational Approach to Commutative Algebra*. Springer-Verlag, New York, 1993.

Dimitri P. Bertsekas. Auction algorithms for network flow problems: A tutorial introduction. *Computational Optimization and Applications*, 1:7–66, 1992.

Navin A. R. Bhat and Kevin Leyton-Brown. Computing Nash equilibria of action-graph games. In *Twentieth Conference on Uncertainty in Artificial Intelligence*, pages 35–42, Banff, 2004.

Sushil Bikhchandani and John W. Mamer. Competitive equilibrium in an exchange economy with indivisibilities. *Journal of Economic Theory*, 74:385–413, 1997.

Ben Blum, Christian R. Shelton, and Daphne Koller. A continuation method for Nash equilibria in structured games. In *Eighteenth International Joint Conference on Artificial Intelligence*, pages 757–764, Acapulco, 2003.

Prosenjit Bose, Anil Maheshwari, and Pat Morin. Fast approximations for sums of distances, clustering, and the Fermat-Weber problem. *Computational Geometry: Theory and Applications*, 24(3):135–146, 2003.

Justin Boyan and Amy Greenwald. Bid determination in simultaneous auctions: An agent architecture. In *Third ACM Conference on Electronic Commerce*, pages 210–212, Tampa, FL, 2001.

John H. Boyd. Symmetries, dynamic equilibria, and the value function. In R. Ramachandran and R. Sato, editors, *Conservation Laws and Symmetry: Applications to Economics and Finance*. Kluwer, Boston, 1990.

Felix Brandt and Gerhard Weiß. Antisocial agents and Vickrey auctions. In *Eighth International Workshop on Agent Theories, Architectures, and Languages*, volume 2333 of *Lecture Notes in Computer Science*, pages 335–347, Seattle, 2001. Springer.

Felix Brandt, Tuomas Sandholm, and Yoav Shoham. Spiteful bidding in sealed-bid auctions. Technical report, Stanford University, 2005.

George W. Brown. Iterative solution of games by fictitious play. In T. C. Koopmans, editor, *Activity Analysis of Production and Allocation*, pages 374–376. Wiley, New York, 1951.

Bruno Buchberger. Gröbner bases: An algorithmic method in polynomial ideal theory. In Nirmal K. Bose, editor, *Multidimensional Systems Theory*. Van Nostrand Reinhold, New York, 1982.

Andrew Byde. Applying evolutionary game theory to auction mechanism design. Technical Report HPL-2002-321, HP Laboratories Bristol, 2002.

Kaylan Chatterjee and W. F. Samuelson. Bargaining under incomplete information. *Operations Research*, 31:835–851, 1983.

Shih-Fen Cheng, Daniel M. Reeves, Yevgeniy Vorobeychik, and Michael P. Wellman. Notes on equilibria in symmetric games. In *AAMAS-04 Workshop on Game Theory and Decision Theory*, New York, 2004.

Shih-Fen Cheng, Evan Leung, Kevin M. Lochner, Kevin O'Malley, Daniel M. Reeves, L. Julian Schvartzman, and Michael P. Wellman. Walverine: A Walrasian trading agent. *Decision Support Systems*, 39:169–184, 2005.

Dave Cliff. Evolving parameter sets for adaptive trading agents in continuous double-auction markets. In *Agents-98 Workshop on Artificial Societies and Computational Markets*, pages 38–47, Minneapolis, MN, 1998.

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

Augustin Cournot. Researches into the mathematical principles of the theory of wealth, 1838. English Edition, ed. N. Bacon (Macmillan, 1897).

Peter Cramton, Yoav Shoham, and Richard Steinberg, editors. *Combinatorial Auctions*. MIT Press, 2005.

Peter Cramton. Money out of thin air: The nationwide narrowband PCS auction. *Journal of Economics and Management Strategy*, 4:267–343, 1995.

Peter Cramton. Simultaneous ascending auctions. In Peter Cramton, Yoav Shoham, and Richard Steinberg, editors, *Combinatorial Auctions*. MIT Press, 2005.

János A. Csirik, Michael L. Littman, Satinder Singh, and Peter Stone. FAucS: An FCC spectrum auction simulator for autonomous bidding agents. In *Second International Workshop on Electronic Commerce*, volume 2232 of *Lecture Notes in Computer Science*, pages 139–151. Springer-Verlag, 2001.

Richard Dawkins. *The Selfish Gene*. Oxford University Press, 1976.

Sven de Vries and Rakesh Vohra. Combinatorial auctions: A survey. *INFORMS Journal on Computing*, 15:284–309, 2003.

Gerard Debreu. A social equilibrium existence theorem. *Proceedings of the National Academy of Sciences*, 38:886–893, 1952.

Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

Daniel Friedman and John Rust, editors. *The Double Auction Market*. Addison-Wesley, 1993.

Daniel Friedman. Evolutionary games in economics. *Econometrica*, 59:637–666, 1991.

Clemens Fritschi and Klaus Dorer. Agent-oriented software engineering for successful tac participation. In *First International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Bologna, Italy, 2002.

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

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

M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, M. Booth, and F. Rossi. *GNU Scientific Library Reference Manual (2nd Ed.)*. ISBN 0954161734, 2002. `http://www.gnu.org/software/gsl/`.

Herbert Gintis. *Game Theory Evolving*. Princeton University Press, 2000.

Michael B. Gordy. Computationally convenient distributional assumptions for common-value auctions. *Computational Economics*, 12:61–78, 1998.

Srihari Govindan and Robert Wilson. Structure theorems for game trees. *Proceedings of the National Academy of Sciences*, 99(13):9077–9080, 2002.

Srihari Govindan and Robert Wilson. A global Newton method to compute Nash equilibria. *Journal of Economic Theory*, 110:65–86, 2003.

Amy Greenwald and Justin Boyan. Bidding under uncertainty: Theory and experiments. In *Twentieth Conference on Uncertainty in Artificial Intelligence*, pages 209–216, Banff, 2004.

Amy Greenwald and Peter Stone. The first international trading agent competition: Autonomous bidding agents. *IEEE Internet Computing*, 5(2):52–60, 2001.

John Harsanyi. Games of incomplete information played by Bayesian players. *Management Science*, 14:159–182, 320–334, 486–502, 1967.

Minghua He and Nicholas R. Jennings. SouthamptonTAC: Designing a successful trading agent. In *Fifteenth European Conference on Artificial Intelligence*, pages 8–12, Lyon, 2002.

Minghua He and Nicholas R. Jennings. SouthamptonTAC: An adaptive autonomous trading agent. *ACM Transactions on Internet Technology*, 3:218–235, 2003.

Shlomit Hon-Snir, Dov Monderer, and Aner Sela. A learning approach to auctions. *Journal of Economic Theory*, 82:65–88, 1998.

John H. Kagel and Dan Levin. Independent private value auctions: Bidder behavior in first-, second-, and third-price auctions with varying numbers of bidders. *Economic Journal*, 103(419):868–878, 1993.

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

Alexander S. Kelso and Vincent P. Crawford. Job matching, coalition formation, and gross substitutes. *Econometrica*, 50:1483–1504, 1982.

Jeffrey O. Kephart and Amy R. Greenwald. Shopbot economics. *Autonomous Agents and Multiagent Systems*, 5:255–287, 2002.

Jeffrey O. Kephart, James E. Hanson, and Jakka Sairamesh. Price and niche wars in a free-market economy of software agents. *Artificial Life*, 4:1–23, 1998.

Daphne Koller and Avi Pfeffer. Representations and solutions for game-theoretic problems. *Artificial Intelligence*, 94(1–2):167–215, 1997.

Daphne Koller, Nimrod Megiddo, and Bernhard von Stengel. Efficient computation of equilibria for extensive two-person games. *Games and Economic Behavior*, 14:247–259, 1996.

David M. Kreps. *Game Theory and Economic Modelling*. Oxford University Press, 1990.

Vijay Krishna and John Morgan. An analysis of the war of attrition and the all-pay auction. *Journal of Economic Theory*, 72:343–362, 1997.

Vijay Krishna. *Auction Theory*. Academic Press, 2002.

Pierfrancesco La Mura. Game networks. In *Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 335–342, Stanford, 2000.

Blake LeBaron. Agent-based computational finance: Suggested readings and early research. *Journal of Economic Dynamics and Control*, 24:679–702, 2000.

Pierre L'Ecuyer. Efficiency improvement and variance reduction. In *Winter Simulation Conference*, pages 122–132, 1994.

Wolfgang Leininger, Peter Linhart, and Roy Radner. Equilibria of the sealed-bid mechanism for bargaining with incomplete information. *Journal of Economic Theory*, 48:63–106, 1989.

Carlton E. Lemke and J. T. Howson, Jr. Equilibrium points of bimatrix games. *Journal of the Society for Industrial and Applied Mathematics*, 12(2):413–423, 1964.

Kevin Leyton-Brown and Moshe Tennenholtz. Local-effect games. In *Eighteenth International Joint Conference on Artificial Intelligence*, pages 772–777, 2003.

Kevin Leyton-Brown, Mark Pearson, and Yoav Shoham. Towards a universal test suite for combinatorial auction algorithms. In *Second ACM Conference on Electronic Commerce*, pages 66–76, 2000.

Richard J. Lipton, Evangelos Markakis, and Aranyak Mehta. Playing large games using simple strategies. In *Fourth ACM Conference on Electronic Commerce*, pages 36–41, San Diego, 2003.

Jeffrey K. MacKie-Mason and Michael P. Wellman. Automated markets and trading agents. In *Handbook of Agent-Based Computational Economics*. Elsevier, 2005.

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

Andreu Mas-Colell, Michael D. Whinston, and Jerry R. Green. *Microeconomic Theory*. Oxford University Press, New York, 1995.

R. Preston McAfee and John McMillan. Auctions and bidding. *Journal of Economic Literature*, 25:699–738, 1987.

R. Preston McAfee and John McMillan. Analyzing the airwaves auction. *Journal of Economic Perspectives*, 10(1):159–175, 1996.

Richard D. McKelvey and Andrew McLennan. Computation of equilibria in finite games. In *Handbook of Computational Economics*, volume 1. Elsevier Science, 1996.

Richard D. McKelvey, Andrew McLennan, and Theodore Turocy. Gambit game theory analysis software and tools, 1992. `http://econweb.tamu.edu/gambit`.

Flavio M. Menezes, Paulo K. Monteiro, and Akram Temimi. Private provision of discrete public goods with incomplete information. *Journal of Mathematical Economics*, 35(4):493–514, July 2001.

Paul R. Milgrom and John Roberts. Adaptive and sophisticated learning in normal form games. *Games and Economic Behavior*, 3:82–100, 1991.

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

Paul R. Milgrom and Robert J. Weber. Distributional strategies for games with incomplete information. *Mathematics of Operations Research*, 10:619–632, 1985.

Paul Milgrom. Putting auction theory to work: The simultaneous ascending auction. *Journal of Political Economy*, 108:245–272, 2000.

John Morgan, Ken Steiglitz, and George Reis. The spite motive and equilibrium behavior in auctions. *Contributions to Economic Analysis and Policy*, 2(1), 2003.

John Nash. *Non-cooperative games*. PhD thesis, Princeton University, Department of Mathematics, 1950.

John Nash. Non-cooperative games. *Annals of Mathematics*, 2(54):286–295, 1951.

John Nash. Two-person cooperative games. *Econometrica*, 21:128–140, 1953.

John A. Nelder and Roger Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

James Nicolaisen, Valentin Petrov, and Leigh Tesfatsion. Market power and efficiency in a computational electricity market with discriminatory double-auction pricing. *IEEE Transactions on Evolutionary Computation*, 5:504–523, 2001.

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

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

Michael Peters and Sergei Severinov. Internet auctions with many traders. Technical report, University of Toronto, 2001.

Steve Phelps, Simon Parsons, Peter McBurney, and Elizabeth Sklar. Co-evolution of auction mechanisms and trading strategies: Towards a novel approach to microeconomic design. In *GECCO-02 Workshop on Evolutionary Computation in Multi-Agent Systems*, pages 65–72, 2002.

Ryan Porter, Eugene Nudelman, and Yoav Shoham. Simple search methods for finding a Nash equilibrium. In *Nineteenth National Conference on Artificial Intelligence*, pages 664–669, San Jose, CA, 2004.

William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C*. Cambridge University Press, second edition, 1992.

Tony Curzon Price. Using co-evolutionary programming to simulate strategic behaviour in markets. *Journal of Evolutionary Economics*, 7:219–254, 1997.

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

Daniel M. Reeves, Michael P. Wellman, Jeffrey K. MacKie-Mason, and Anna Osepayshvili. Exploring bidding strategies for market-based scheduling. *Decision Support Systems*, 39(1):67–85, March 2005.

John A. Rice. *Mathematical Statistics and Data Analysis*. Duxbury Press, California, 1995.

Julia Bowman Robinson. An iterative method of solving a game. *Annals of Mathematics*, 54:296–301, 1951.

Jeffrey S. Rosenschein and Gilad Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. MIT Press, 1994.

Sheldon M. Ross. *Simulation (3rd Ed.)*. Academic Press, 2001.

John Rust, John Miller, and Richard Palmer. Behavior of trading automata in a computerized double auction market. In Friedman and Rust [1993], pages 155–198.

Mark A. Satterthwaite and Steven R. Williams. Bilateral trade with the sealed bid k-double auction: Existence and efficiency. *Journal of Economic Theory*, 48:107–133, 1989.

Mark A. Satterthwaite and Steven R. Williams. The Bayesian theory of the k-double auction. In Friedman and Rust [1993], pages 99–123.

Peter Schuster and Karl Sigmund. Replicator dynamics. *Journal of Theoretical Biology*, 100:533–538, 1983.

Reinhard Selten and Joachim Buchta. Experimental sealed bid first price auctions with directly observed bid functions. Technical Report Discussion Paper B-270, University of Bonn, 1994.

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

Lloyd S. Shapley. Some topics in two-person games. In M. Dresher, L. S. Shapley, and A. W. Tucker, editors, *Advances in Game Theory*, volume 5 of *Annals of Mathematical Studies*, pages 1–28. 1964.

Satinder Singh, Vishal Soni, and Michael P. Wellman. Computing approximate Bayes-Nash equilibria in tree-games of incomplete information. In *Fifth ACM Conference on Electronic Commerce*, pages 81–90, New York, 2004.

Peter Stone and Amy Greenwald. The first international trading agent competition: Autonomous bidding agents. *Journal of Electronic Commerce Research*, 5:229–265, 2005.

Peter Stone, Michael L. Littman, Satinder Singh, and Michael Kearns. ATTac-2000: An adaptive autonomous bidding agent. *Journal of Artificial Intelligence Research*, 15:189–206, 2001.

Peter Stone, Robert E. Schapire, Jánosz A. Csirik, Michael L. Littman, and David McAllester. ATTac-2001: A learning, autonomous bidding agent. In *Agent-Mediated Electronic Commerce IV*, volume 2153 of *Lecture Notes in Computer Science*. Springer-Verlag, 2002.

Peter Stone, Robert E. Schapire, Michael L. Littman, János A. Csirik, and David McAllester. Decision-theoretic bidding based on learned density models in simultaneous, interacting auctions. *Journal of Artificial Intelligence Research*, 19:209–242, 2003.

Peter Stone. Multiagent competitions and research: Lessons from RoboCup and TAC. In *Sixth RoboCup International Symposium*, Fukuoka, Japan, 2002.

Peter D. Taylor and Leo B. Jonker. Evolutionary stable strategies and game dynamics. *Mathematical Biosciences*, 40:145–156, 1978.

Theodore L. Turocy. *Computation and Robustness in Sealed-bid Auctions*. PhD thesis, Northwestern University, 2001.

Stanislav Uryasev and Reuven Y. Rubinstein. On relaxation algorithms in computation of non-cooperative equilibria. In *IEEE Transactions on Automatic Control (Technical Notes)*, volume 39, 1994.

Eric van Damme. Refinements of the Nash equilibrium concept. In *Lecture Notes in Economics and Mathematical Systems*. Springer, Berlin and New York, 1983.

Ioannis A. Vetsikas and Bart Selman. A principled study of the design tradeoffs for autonomous trading agents. In *Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 473–480, Melbourne, 2003.

William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1947.

Léon Walras. Elements of pure economics, 1874. English translation by William Jaffé, published by Allen and Unwin, 1954.

William E. Walsh, Michael P. Wellman, and Fredrik Ygge. Combinatorial auctions for supply chain formation. In *Second ACM Conference on Electronic Commerce*, pages 260–269, 2000.

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

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

William E. Walsh. *Market Protocols for Decentralized Supply Chain Formation*. PhD thesis, University of Michigan, 2001.

Robert J. Weber. Making more from less: Strategic demand reduction in the FCC spectrum auctions. *Journal of Economics and Management Strategy*, 6:529–548, 1997.

Jörgen W. Weibull. *Evolutionary Game Theory*. MIT Press, 1995.

Michael P. Wellman, William E. Walsh, Peter R. Wurman, and Jeffrey K. MacKie-Mason. Auction protocols for decentralized scheduling. *Games and Economic Behavior*, 35:271–303, 2001.

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

Michael P. Wellman, Shih-Fen Cheng, Daniel M. Reeves, and Kevin M. Lochner. Trading agents competing: Performance, progress, and market effectiveness. *IEEE Intelligent Systems*, 18(6):48–53, 2003.

Michael P. Wellman, Amy Greenwald, Peter Stone, and Peter R. Wurman. The 2001 trading agent competition. *Electronic Markets*, 13:4–12, 2003.

Michael P. Wellman, Jeffrey K. MacKie-Mason, Daniel M. Reeves, and Sowmya Swaminathan. Exploring bidding strategies for market-based scheduling. In *Fourth ACM Conference on Electronic Commerce*, pages 115–124, San Diego, June 2003.

Michael P. Wellman, Daniel M. Reeves, Kevin M. Lochner, and Yevgeniy Vorobeychik. Price prediction in a trading agent competition. *Journal of Artificial Intelligence Research*, 21:19–36, 2004.

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

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

Michael P. Wellman, Daniel M. Reeves, Kevin M. Lochner, and Rahul Suri. Searching for Walverine 2005. In *IJCAI-05 Workshop on Trading Agent Design and Analysis*, Edinburgh, 2005.