Mechanism Design and Analysis Using Empirical Game Models

Yevgeniy Vorobeychik

University of Michigan
Artificial Intelligence Laboratory
1101 Beal Ave.
Ann Arbor, MI 48109-2110, USA
yvorobey@umich.edu

January 9, 2007

1 Introduction

The mechanism design problem has been a major subject of applied game-theoretic research for many years now. Commonly, it is approached theoretically from the perspective of designing rules that ensure that truth revelation is a Bayes-Nash equilibrium strategy or a dominant strategy for agents, and the outcomes maximize the designer's objective [Mas-Colell et al., 1995, Fudenberg and Tirole, 1991, Osborne and Rubinstein, 1994]. This approach is, in principle, very general, as attested to by the *revelation principle* [Mas-Colell et al., 1995]. However, it is difficult to imagine real policy-making decisions that involve querying every affected party for their complete preferences. Intuitively, we can recognize that typically the domain over which individual preferences must be specified is intractably large or infinite, and the problem of preference elicitation may be further complicated by psychological and socialogical factors that prevent people from recognizing their *true* preferences. These difficulties confound direct revelation mechanisms in all but relatively simple domains. In practice, we must attempt to tackle the full complexity of indirect mechanism design, as even reasonable heuristics may go a long way to address real policy-making problems.

On another dimension of mechanism design lies system modeling. Game theorists have been building analytically tractable models of relevant phenomena for many years, attempting to glean understanding of general system dynamics based on relatively simple and clean analytical solutions. Such methods have indeed enjoyed tremendous success, as they simplify the problems down to relevant dynamics, allowing clear and compelling explanations of sometimes extremely complex phenomena. However, a natural shortcoming of analytically tractable modeling is, well, the analytical tractability constraint. While the simple models may contain tremendous latent power, we may frequently require a more complex class of models which, while not analytically tractable, models the system of interst more accurately. The acuteness of this need has been demonstrated by several problems of recent interest: matching markets for American physicians and FCC frequency spectrum auctions [Roth, 2002]. In both of these domains, the shortcomings of analytic simplifications have become especially apparent to the designers.

The complication of expanding the class of "allowable models" beyond analytical tractability is the shortage of tools that allow us to analyze such models. Particularly, while there are many numerical techniques for analyzing games defined with finite payoff matrices, no general tools exist for infinite game analysis, and even large finite games are in practice intractable. There are also a number of techniques for approximate analysis of intractable games, but these are typically tied to applications [Reeves et al., 2005, Wellman et al., 2005a,b] and none share an interface. Furthermore, if payoffs are only available via noisy realizations, probabilistic analysis is needed to evaluate how much a game defined by noisy samples from the actual payoff functions reflects the actual strategic scenario.

The contributions I propose are in the form of methods and asymptotic convergence results, as well as applications to particular strategic scenarios and mechanism design problems of interest. In "Empirical Mechanism Design: Methods, with an Application to a Supply Chain Scenario" Vorobeychik et al. [2005a], I, along with several co-authors, present methods for mechanism design in the case where the payoff function is specified by a simulator or, more generally, a "black box". We apply these methods to a particular mechanism design problem that has interested a group of researchers in the Trading Agent Competition community for the last several years. We also present some asymptotic convergence guarantees for approximately optimal mechanisms.

In my work, I hope to address two problems fundamental to the empirical mechanism design setting (and, perhaps, to mechanism design in general). Both of these problems stem from the issue of evaluating the designer's utility for a given mechanism choice.

The first problem is that of determining a set of solutions to a game induced by a particular mechanism choice, our presumption being that the designer evaluates his utility with respect to these solutions. Since the empirical mechanism design framework incorporates empirical games, rather than fully specified games, as a part of analysis, there is a need for a tool that can provide solutions or estimates of solutions based on empirical games. To establish a unified framework for analysis of empirical games—that is, games that have a black-box specification of the players' payoff functions—I suggest an Empirical Game Analysis Toolkit. Such a toolkit would provide an analyst with a uniform interface to all analysis tools by incorporating the available numerical techniques and using an appropriate generalization of input and output.

In "Learning Payoff Functions in Infinite Games" Vorobeychik et al. [2005b], I present a particular set of tools for analyzing infinite games for which payoff data is only available in the form of a noisy data set. In what follows, I propose a number of additional methods that estimate a sample solution or a set of solutions to a game specified using a "black box" payoff function (such as a data set of payoffs, for example). Furthermore, I derive in this work probabilistic bounds in order to assess how close the solution estimates based on empirical games are to actual solutions. To my knowledge, this is the first analytic derivation of such bounds for the setting of games with noisy payoffs.

The second significant problem in mechanism design is that of formulating sensible beliefs over the set of solutions for a particular strategic scenario induced by a mechanism, I intend to explore various ways such beliefs could be formed. To this end, I introduce solution concepts or generalizations of well-known solution concepts which, I believe, will aid analysis in realistic design settings.

My applications are not merely limited to a mechanism design setting per se. It is important to recognize that empirical game analysis methods are significant in themselves, as a policy-maker may be interested in assessing the effect of currently implemented mechanism before he would endeavor to revise it.

Thus, I have collaborated in past work in applying the empirical game analysis methods described here in several domains that are not all directly related to mechanism design. Several applications have been in the context of strategic analysis of the Trading Agent Competition. Another nearly completed project anlayzes strategic interactions in combat in which outcomes are obtained using simulations. In the completed part of the project I model the strategic interactions between combatants as a zero-sum game. In the next part of the project, I expect to extend this model to general-sum games. Finally, I hope to apply my methods both in strategic analysis and indirect mechanism design in order to design optimal auctions in analytically intractable domains.

2 Preliminaries

2.1 Notation

A *normal form game*[1] is formally expressed as $[I, \{R_i\}, \{u_i(r)\}]$, where I refers to the set of players and $m = |I|$ is the number of players. R_i is the set of strategies available to player $i \in I$, with $R = R_1 \times \ldots \times R_m$ representing the set of joint strategies of all players. I designate the set of pure strategies available to player i by A_i, and denote the joint set of pure strategies of all players by $A = A_1 \times \ldots \times A_m$. It is often convenient to refer to a strategy of player i separately from that of the remaining players. To accommodate this, I use a_{-i} to denote the joint strategy of all players other than player i.

Let S_i be the set of all probability distributions (mixtures) over A_i and, similarly, S be the set of all distributions over A. An $s \in S$ is called a *mixed strategy profile*. When the game is finite (i.e., A and I are both finite), the probability that $a \in A$ is played under s is written $s(a) = s(a_i, a_{-i})$. When the distribution s is not correlated, we can simply say $s_i(a_i)$ when referring to the probability player i plays a_i under s.

Next, I define the payoff (utility) function of each player i by $u_i : A_1 \times \cdots \times A_m \to \mathbb{R}$, where $u_i(a_i, a_{-i})$ indicates the payoff to player i to playing pure strategy a_i when the remaining players play a_{-i}. We can extend this definition to mixed strategies by assuming that u_i are von Neumann-Morgenstern (vNM) utilities as follows: $u_i(s) = E_s[u_i(a)]$, where E_s is the expectation taken with respect to the probability distribution of play induced by the players' mixed strategy s. When the game has finitely many pure strategies, this definition is equivalent to $u_i(s) = \sum_{a \in A} u_i(a)s(a)$.

Occasionally, I write $u_i(x, y)$ to mean that $x \in A_i$ or S_i and $y \in A_{-i}$ or S_{-i} depending on context. I also express the set of utility functions of all players as $u(\cdot) = \{u_1(\cdot), \ldots, u_m(\cdot)\}$.

I define a function, $\epsilon : R \to \mathbb{R}$, interpreted as the maximum benefit any player can obtain by deviating from its strategy in the specified profile.

$$\epsilon(r) = \max_{i \in I} \max_{a_i \in A_i} [u_i(a_i, r_{-i}) - u_i(r)], \tag{1}$$

where r belongs to some strategy set, R, of either pure or mixed strategies.

2.2 Nash Equilibrium and Symmetry

Faced with a game, an agent would ideally play its best strategy given those played by the other agents. A configuration where all agents play strategies that are best responses to the others

[1] By employing the normal form, we model agents as playing a single action, with decisions taken simultaneously. This is appropriate for the current study, which treats strategies (agent programs) as atomic actions. We could capture finer-grained decisions about action over time in the *extensive form*. Although any extensive game can be recast in normal form, doing so may sacrifice compactness and blur relevant distinctions (e.g., subgame perfection).

constitutes a *Nash equilibrium.*

Definition 1. *A strategy profile* $r = (r_1, \ldots, r_m)$ *constitutes a* Nash equilibrium *of game* $[I, \{R_i\}, \{u_i(r)\}]$ *if for every* $i \in I$, $r'_i \in R_i$, $u_i(r_i, r_{-i}) \geq u_i(r'_i, r_{-i})$.

When $r \in A$, the above defines a *pure strategy Nash equilibrium*; otherwise the definition describes a *mixed strategy Nash equilibrium.* I often appeal to the concept of an *approximate*, or ϵ*-Nash equilibrium*, where ϵ is the maximum benefit to any agent for deviating from the prescribed strategy. Thus, $\epsilon(r)$ as defined above (1) is such that profile r is an ϵ-Nash equilibrium iff $\epsilon(r) \leq \epsilon$.

In this study I devote particular attention to games that exhibit symmetry with respect to payoffs, rendering agents strategically identical.

Definition 2. *A game* $[I, \{R_i\}, \{u_i(r)\}]$ *is* symmetric *if* $\forall i, j \in I$,

- $R_i = R_j$, *and*
- $u_i(r_i, r_{-i}) = u_j(r_j, r_{-j})$ *whenever* $r_i = r_j$ *and* $r_{-i} = r_{-j}$.

Symmetric games have relatively compact descriptions and may present associated computational advantages [Cheng et al., 2004]. Given a symmetric game, we may focus on the subclass of symmetric equilibria, which are arguably most natural [Kreps, 1990], and avoid the need to coordinate on roles.[2] In fairly general settings, symmetric games do possess symmetric equilibria [Nash, 1951, Cheng et al., 2004].

2.3 Empirical Games

In this work, I direct much of my attention to the analysis of *empirical games*, that is games, for which the set of players and their strategy sets are specified, but we do not have a direct specification of the utility (payoff) functions. What we have instead is a partial or inexact specification of the payoff functions of all players using a set of data points.

In formal notation, the empirical game is $[I, R, \mathcal{D}]$, where, I is the set of players and R is the set of joint strategies. $\mathcal{D}$ is a set of data points $\{d_1, \ldots, d_L\}$, where each data point $d_j = (a, V)$ is composed of a particular pure strategy profile, $a \in A$, together with associated characteristics, denoted by V, that are relevant to the specific empirical game. A simple example of a data point is $d_j = (a, U)$, where U is the vector of realized payoffs. A more complicated example could be $d_j = (a, U_1, \ldots, U_K)$, where each U_k is a vector of realized payoffs sampled from a payoff function with noise. Finally, we could have $d_j = (a, M_1, \ldots, M_K)$, where M_k is a statistic of the data set.

If every pure strategy profile in R[3] contains a corresponding data point in $\mathcal{D}$, I say that the data set is *complete.* In this case, we can use the empirical game to estimate the underlying game by $[I, R, \{\bar{u}(r)\}]$, where $\bar{u}(a)$ is the sample average of all U that correspond to a in the data set, and $\bar{u}(r)$ is then derived appropriately as described above (taking it to be a vNM utility function). More generally, we can drop the requirement that the data set is complete, and abstractly use some approximate payoff functions, $\hat{u}(a)$ based on the data to obtain an approximate game, $[I, R, \{\hat{u}(r)\}]$.

[2]Contention may arise when there are disparities among payoffs in asymmetric equilibrium. Even for symmetric equilibria, coordination issues may still be present with respect to equilibrium selection.

[3]For example, if R is the set of joint mixed strategies, the subset of pure strategies would be those $r \in R$ that have *support* of size 1, or, alternatively, that place positive probability on exactly 1 strategy.

3 Empirical Mechanism Design Framework

I distill the problem of mechanism design as follows. Let us imagine that a designer has a parameter, $\theta \in \Theta$, the settings of which determine the designer's control over the mechanism. Θ will thereby define the space of feasible mechanisms. It may be that the designer selects from several possible auctions he can run to sell a particular item, in which case each of the auctions can be associated with some θ. Or the designer may have decided on a particular auction type, modulo some parameter setting. For example, the designer may have decided to run an open-bid auction, but instead of revealing bids exactly, wants to add mean-zero normal noise to each bid. In that case, the mechanism design setting is restricted to choosing the variance of the noise to be added to bids.

The mechanism design problem has received considerable attention in the literature. Indeed, the problem of designing optimal auctions alone has been address in numerous contexts, including [Myerson, 1981, Riley and Samuelson, 1981, Krishna, 2002]. What distinguishes my approach from much of the traditional literature is that I consider a space of empirical games rather than well-specified and analytically tractable games as the source of mechanism outcomes. As I have already suggested, this extends the set of possible strategic models, but also introduces important imediments to analysis: numerical tools are now required to solve or approximate solutions to empirical games.

The designer's ignorance of the players' payoff functions also introduces some philosophical difficulties into my framework. The presumption is that, while the mechanism designer acts without knowledge of the game and is, therefore, ignorant of its solutions, all the players, in their infinite wisdom, play an actual solution (say, Nash equilibrium). A justification of such a view is that the actual strategic setting is extremely complex, and the model, even based on simulations, is still a gross simplification. After a design is chosen, the actual induced agent play will, on aggregate, be similar to a solution. Perhaps a slightly more compelling view, and one that has also had a long tradition of game-theoretic research, is that agents ultimately *learn* to play nearly rationally. Indeed, a broad theoretical and empirical literature demonstrates convergence of learning to Nash equilibria in a number of special contexts [Milgrom and Roberts, 1991, Kalai and Lehrer, 1993, Fudenberg and Levine, 1998]. However, this assumption is somewhat dubious in its own right, since no general convergent learning algorithm is yet known. In order to alleviate these philosophical problems, I use the term "solutions" rather then refering to Nash equilibria, which are specific solution concepts. Later, I suggest several other solution concepts that attempt to incorporate indirectly the concepts of bounded rational agents without specifically modeling bounded rationality, as was done by Rubinstein [1997].

A formal framework for empirical mechanism design is presented in [Vorobeychik et al., 2005a]. Here, I summarize several important points. We modeled the strategic interactions between the designer of the mechanism and its participants as a two-stage game. The designer moves first by selecting a value, θ, from a set of allowable mechanism settings, Θ. All the participant agents observe the mechanism parameter θ and move simultaneously thereafter. For example, the designer could be deciding between a first-price and second-price sealed-bid auction mechanisms, with the presumption that after the choice has been made, the bidders will participate with full awareness of the auction rules.

Since the participants play with full knowledge of the mechanism parameter, we defined a game between them in the second stage as

$$\Gamma_\theta = [I, \{R_i\}, \{u_i(r, \theta)\}].$$

Let $\mathcal{N}(\theta)$ be the set of strategy profiles considered *solutions* of the game Γ_θ.[4]

Suppose that the goal of the designer is to optimize the value of some welfare function, $W(r, \theta)$, dependent on the mechanism parameter and resulting play, r. We defined a pessimistic measure, $W(\hat{R}, \theta) = \inf\{W(r, \theta) : r \in \hat{R}\}$, representing the worst-case welfare of the game induced by θ, assuming that agents play some joint strategy in $\hat{R}$. Typically we care about $W(\mathcal{N}(\theta), \theta)$, the worst-case outcome of playing *some* solution.[5]

We have attempted to tackle this problem in the TAC/SCM domain by searching for a setting of storage cost that would curb total initial procurement of supplier components by the manufacturing agents ([Kiekintveld et al., 2005],[Vorobeychik et al., 2005a]).[6] During the 2004 tournament, the designers of the supply-chain game chose to dramatically increase storage costs as a measure aimed at reducing day-0 procurement, to little avail. In [Vorobeychik et al., 2005a], we systematically explored the relationship between storage costs and the aggregate quantity of components procured on day 0 in equilibrium. We modeled the designer's welfare function as a threshold on the sum of day-0 purchases. Let $\phi(a) = \sum_{i=1}^{6} a_i$ be the aggregation function representing the sum of day-0 procurement of the six agents participating in a particular supply-chain game (for mixed strategy profiles s, I take expectation of ϕ with respect to the mixture). The designer's welfare function is then given by $W(\mathcal{N}(\theta), \theta) = \mathbf{I}\{\sup\{\phi(\mathcal{N}(\theta))\} \leq \alpha\}$, where α is the maximum acceptable level of day-0 procurement and $\mathbf{I}$ is the indicator function. The designer selects a value θ of storage costs, expressed as an annual percentage of the baseline value of components in the inventory (charged daily), from the set $\Theta = \mathbb{R}^+$. The data was collected by simulating TAC/SCM games on a local version of the 2004 TAC/SCM server, which has a configuration setting for the storage cost.

4 Convergence of Optimal Design Based on Empirical Data

In [Vorobeychik et al., 2005a] several co-authors and I prove that under certain restrictions which include finite player and strategy sets, as well as finite set of design parameter choices, optimal design parameter based on games with noisy payoff data will almost surely converge to a true optimal design choice. Additionally, we provide some theoretical justification for using ϵ- Nash equilibria as estimates of actual Nash equilibria (that is, equilibria of the underlying game): for any positive ϵ, when enough samples are taken, the set of ϵ-Nash equilibria of an empirical game contains all actual Nash equilibria. A form of the converse is true as well: eventually, every Nash equilibrium of an empirical game will be arbitrarily close (in any metric space) to some Nash equilibrium of the underlying game.

[4] We generally adopt Nash equilibrium as the solution concept, and thus take $\mathcal{N}(\theta)$ to be the set of equilibria. However, much of the methodology developed here could be employed with alternative criteria for deriving agent behavior from a game definition.

[5] Again, alternatives are available. For example, if one has a probability distribution over the solution set $\mathcal{N}(\theta)$, it would be natural to take the expectation of $W(r, \theta)$ instead. I will discuss this explicitly below.

[6] The motivation for this problem arose as a result of TAC/SCM experience in the 2003/2004 tournaments.

5 Strategic Modeling with Empirical Games

At the core of empirical mechanism design problems lies the problem of strategic modeling using empirical games. First of all, the designer will need an estimate of the set of solutions to a game induced by a mechanism choice. Additionally, I can envision many other settings, which would not formally fall under the rubric of mechanism design, but which nevertheless provide motivation to undertake strategic analysis in an empirical game setting. Many real mechanism design problems may be difficult to formulate, and the formulation itself may depend in complex ways on evaluating a design choice already in place. This is commonly true of political processes. Before proposing healthcare or tax reforms, politicians must first provide evidence that current systems are sufficiently ineffective to warrant such reforms, and the more complex the proposed changes, the more "broken" the current system must be to justify them.

Naturally, game theoretic literature has been effectively designing and solving strategic interaction scenarios for many years now. However, design effort typically concentrates on finding analytic means to dissect the relevant situation, either through explicit modeling of player utility functions, or endowing them with general properties and doing some comparative statics [Mas-Colell et al., 1995]. In order to remain analytically tractable, any such methods must make numerous simplifying assumptions, often ignoring relevant characteristics of the problem. While these methods have allowed us to gain much theoretical ground in understanding and predicting real strategic scenarios, their shortcoming manifests themselves particularly strongly when economists take up the role of system engineers [Roth, 2002]. For an engineer, the details that would be abstracted away in standard theory may well be of utmost importance. But accounting for such details often confounds the analysis, and, consequently, engineers of economic systems must employ additional methods, usually ad hoc, and often simulation-based, in order to inform ultimate decisions. For example, Roth and Peranson [1999], in their efforts to design the matching market for American physicians, found that past theoretical results were impractical precisely because the more complex reality failed to satisfy many of the underlying assumptions in the theory. Instead, they used the theory as a good starting point for algorithmic design, but complemented it with a series of computational experiments.

Another famous economic engineering design example is the design of FCC Spectrum auctions [McMillan, 1994, Milgrom, 2000, Roth, 2002]. Even in the light of considerable literature in optimal auction design [Myerson, 1981, Riley and Samuelson, 1981, Krishna, 2002], several attempts at auction design for spectrum licenses had proven to be flawed [McMillan, 1994, Roth, 2002]. The numerous complementarities, information asymmetries, as well as the complexity of auction design goals made this an extremely difficult problem in practice, and many alternative auction designs and subsequent revisions were required to find a workable design [Roth, 2002].

My goal in this work is to address such problems of economic engineering head on and in a principled way. Traditional game theory relies on solutions (e.g., Nash equilibria) of analytical models. In more complex scenarios, such models may be too simplistic. In principle, we can build a considerably more complex model as a computer program that outputs a value of players' utilities given their joint strategy profile (and given joint type realizations when relevant). Such a program is a *simulator* of payoffs, and we can envision running multiple *simulations*, i.e., iterations of this program, to obtain payoff realizations for a set of joint strategy profiles of the players. As a result of simulations, we would obtain an *empirical game* as defined in Section 2.3.

In a sense, this simulator serves the purpose of the players' payoff functions. However, it is far from clear how solutions can be obtained given such a description of the players' utilities. This question is addressed later, when I delve into numerical methods for approximating solutions to games. For now, I have yet to finish the discussion of modeling games using such simulators.

The simplest setting is that of simultaneous play, when each query from the simulator can be interpreted as an entry in the normal-form game matrix. Empirical analysis based on such models was done, for example, by Reeves et al. [2005] and Wellman et al. [2005a]. However, observe that, just like in typical game modeling done by game theorists, we have room for endowing our games with dynamic structure. Now, it is well-known that the normal-form game structure is general in a sense that you can mold any given game structure into a normal form by defining strategies appropriately (for example, temporal dependence would be reflected by making strategies functions of histories). However, extensive form captures temporal structure more efficiently and, additionally, allows for solutions besides Nash equilibria (e.g., subgame perfection). Often, extensive form models will also be far more intuitive than the alternative normal-form games when temporal elements are especially important for analysis.

In ongoing work with RAND Corporation [Vorobeychik and Porche, 2005] we have explored a problem in which each agent's strategy is a vector of strategic parameters. These can be set simultaneously for all agents, to effect a simultaneous game, or agents may alternate setting some of their strategic parameters, observing some or all others that have been set in past stages, which would result in an extensive-form game. We have explored both alternatives, developing a simple sequential game model, in which a designated agent moves first by setting a specific strategic parameter, followed in the second stage by the simultaneous play of all the agents.

Since I am suggesting modeling of generic game forms, I must find an encoding which would allow me to use simulations to produce payoffs for an arbitrary game model. Alas, this is an open question still, and I hope to address it in my work in the not far-away future.

6 Empirical Game Analysis Toolkit (EGAT)

Strategic modeling can hardly be effective if the tools necessary to analyze the models are scarce. Since I am proposing a new modeling framework based on empirical games, I need to present an analyst with a set of numerical tools that can elicit the desired solutions based on his models.

In this section, I present the idea of a toolkit that provides a mechanism designer with a set of numerical tools for empirical game analysis. In Figure 1 I depict on a high level how such a toolkit would be used in modeling and analysis of strategic interactions. As his first order of business, the analyst must develop a model of strategic interactions that are of interest to him. If the game defined by the model is very large (many agents and/or very large strategy sets), it will need to be restricted appropriately, ensuring that the model is still well represented by the restricted game. If the simulator is developed by the anlayst, it will naturally be an integral part of the model, since it essentially defines the players' payoff functions. The output of the simulator will be a data set of strategy profiles of and corresponding payoffs to all players, and this will provide the input (along with necessary parameters) to the proposed toolkit. The output of the toolkit would then be a set of desired solutions or approximate solutions, upon which further analysis of the model could anchor. As a way of allowing more general sources of input that could be available to an analyst, we can make the entire simulation and modeling element of the analysis exogenous, as represented by the dotted box. We may even consider a case which has an exogenous data source, but an endogenously constructed model which affects the solutions.

The simplest modeling scenario that I can envision is a game with few players and few strategies

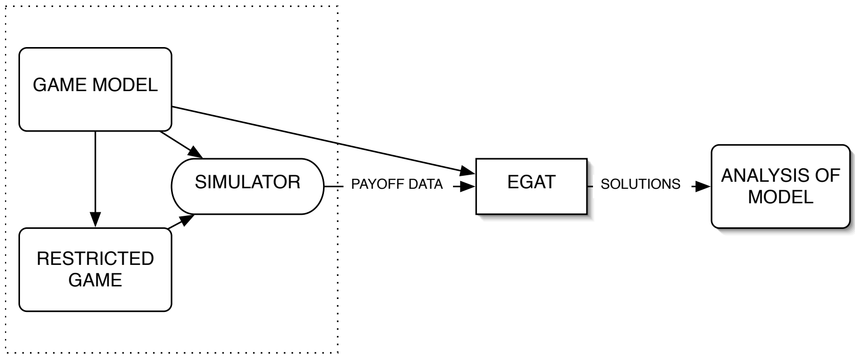

Figure 1: Empirical Game Analysis Toolkit functionality.

for which we can obtain an exact payoff matrix using simulations. Let us say that in such a context, we have access (typically, through simulations) to the entire finite payoff matrix of the game. Then, finite game analysis merely involves solving the finite game numerically. If we find ourselves in such a setting, we can exploit one of numerous finite game solving tools. The most well-known and most general is GAMBIT [McKelvey et al., 2005], which provides tools to find Nash equilibria for normal-form and extensive-form games. Reeves et al. [2005] used Replicator Dynamics [Friedman, 1991] as an alternative method for finding sample Nash equilibria that, in their experience, was more effective than GAMBIT, as it was able to take advantage of symmetry in the game. A recent path following algorithm by Govindan and Wilson [2003] empirically shows improvement over the classical simplicial subdivision [McKelvey and McLennan, 1996] and was implemented as a part of the GameTracer software for finding Nash equilibria in normal-form games [Blume et al., 2003]. Govindan and Wilson also developed an analogous algorithm for extensive-form games [Govindan and Wilson, 2002].

Even more recently, Porter et al. [2006, to appear] present a series of effective search methods for finding a sample Nash equilibrium in an arbitrary normal-form game, while Herings and Peeters [2005] present a convergent algorithm for finding all Nash equilibria in arbitrary finite games. Additionally, a recent work by Sandholm et al. [2005] presents a Mixed-Integer Programming formulation to find a sample or the entire set of Nash equilibria in finite two-person games.

A somewhat more complex scenario has again the same game produced using simulations, but now payoffs produced by the simulator contain noise. We can still use this game as input into EGAT, which would then solve it as if it were the actual game. However, now the output may include probability that the Nash equilibria of the noisy game are actual Nash equilibria. This would require variances as well as numbers of samples taken for each profile to be specified as a part of the empirical game.

If the game is large and especially if it is infinite, it is no longer feasible to use simulations to produce estimates of payoffs for every strategy profile, nor is it generally feasible to solve a game even if the payoff function were available.[7] Instead, our data set will inherently be a partial specification

[7]The lone generic numerical infinite game solver that I am aware of is by Reeves and Wellman [2004], and it works

of the payoff function (matrix). We can define a restricted strategy set and sample the simulator for the payoff matrix of this restricted game. The restricted game can then be fed into EGAT to produce equilibrium estimates for the actual game. In order to obtain the probability distribution over Nash equilibria in this case, we need, in addition to variance information, additional assumptions about, for example, a bound on slope of the payoff function. This could be another input into EGAT.

Another way to curb the complexity of simulating and solving a particular game, the hierarchical player reduction technique, introduced by Wellman et al. [2005b], restricts subsets of players to play the same strategy in any profile that is sampled. By defining equivalence classes with respect to players that are restricted to play the same strategies, a new exponentially smaller game can be defined, in which players are the equivalence classes of the original game. Such a technique could be incorporated into EGAT in several ways. The simplest would be to feed the game *after* the reduction with respect to players had already taken place. As far as EGAT is concerned, it is just another empirical game and the fact that it had been reduced from another should have no implications for solutions. Alternatively, original data set could be used as input, but we could specify also that a data set has a restricted form with respect to subsets of players, and the appropriate reduction should be applied. Finally, we could scan the data set automatically, seeking out equivalence classes of players and applying reductions when appropriate, and merely provide a user with a binary parameter that turns this preprocessor on or off.

In Figure 2 I depict my vision for the structure of the proposed Empirical Game Analysis Toolkit. As mentioned above, input could be fed through a preprocessor that performs hierarchical

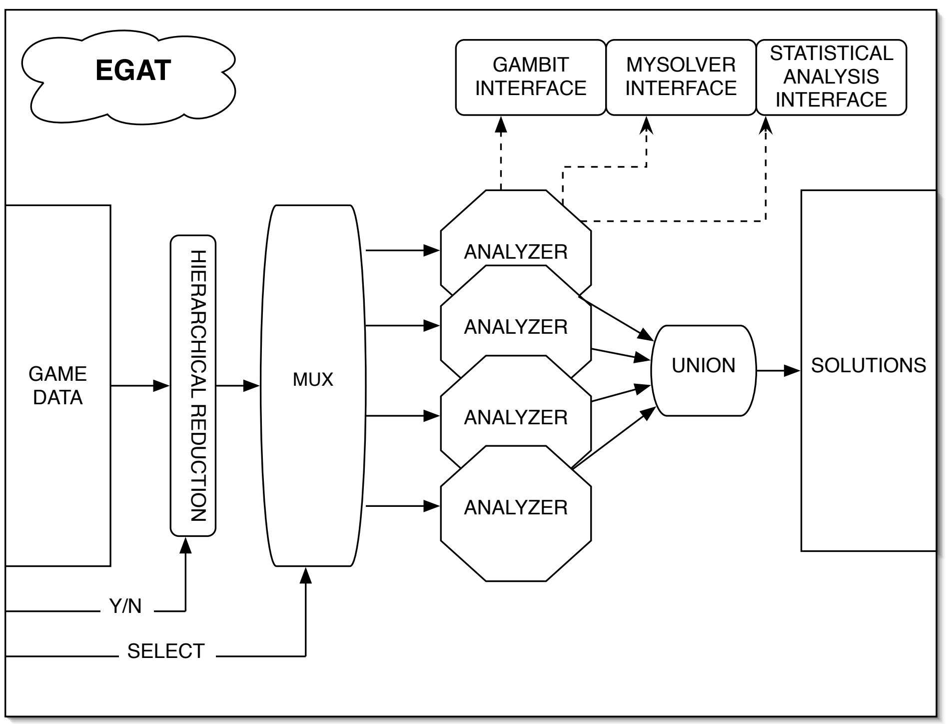

Figure 2: Empirical Game Analysis Toolkit functionality.

reduction, if the option is turned on. It is then fed to a subset of analysis tools (analyzers) that can

by iterative best-response dynimic, which is not guaranteed to converge.

approximate a set of desired solutions (which solutions are desired would have to specified as well). These would be combined in an appropriate way and formatted for output. Each analyzer could in turn rely on interfaces to standard game solvers or custom-made solvers and analysis libraries, that are provided as a part of EGAT distribution.

An important aspect of game analysis may entail interleaving simulations and analysis in order to avoid unnecessary simulation effort. This would require the output of EGAT to be the next set of points to be queried, rather than a set of solutions. For example, the output could be the neighbors of candidate equilibria (defined by one-player devations not present in data set), or some other profiles that need to be sampled (or require additional samples) to improve the quality of equilibrium approximation. Since I keep the "type" of solutions general, and allow the user to specify this type, it should not be difficult to extend EGAT to accomodate a simulation loop, although the interface would need to be extended to provide for solutions such as Nash equilibria to be returned *eventually*.

In the remainder of this section, I describe the input and output interface of EGAT that several collaborators and myself envision for the short term. Then, in sections that follow, I present my work towards developing tools that I have developed and plan to develop to analyze empirical games. Eventually, these will be incorporated into the toolkit as the *analyzer* modules and libraries.

6.1 Input Options

The input interface to EGAT is composed of two parts: *data* and *directions*. The data set will provide a set of samples from a payoff function, whereas the directions file will include the information about the players and their strategy sets. Additionally, the *directions* interface will allow the user to specify which analysis tools are to be used and to provide the instructions for these. Finally, output format would be specified through this interface. For example, in Figure 2, the user will specify whether or not to employ the hierarchical reduction and then select which analyzer will be called on the resulting data set (designated by the schematic representation of a multiplexer).

While EGAT will in the end support multiple input formats, and its architecture will allow adding custom input specifications, a relatively general logical form of the data that can serve as input to EGAT is $(a, t, u(a, t), n(a, t), var(a, t))$, where a is a pure strategy profile, t is the profile of player types (as a special case, a game of complete information will have a fixed t for every player; furthermore, if anlyzers are called that only process games of complete information, the type entry would be ignored), and $u(a, t)$ is the payoff vector to all players when players with types in t play a. The payoff value may be sum of payoffs from $n(a, t)$ noisy samples, or a single exact payoff (in which case $n(a, t)$ will be 1). $var(a, t)$ is the variance of noisy samples, and it will, naturally, be 0 if samples contain no noise.[8]

While the implementation issues may need to be transparent for individual solvers, they may well be important when answers to queries are needed in terms of specific strategic attributes. Consequently, in addition to the actual data, the input interface will specify the data types of each of the above data entries. Each pure strategy and player's data type is allowed to be a vector of three possible types: integer, real number, or a string. The payoff entry is restricted to be an integer or a string, as is variance. Finally, the number of samples is naturally restricted to be an integer.

[8]For computational reasons, the actual implementation will use the sum of squares instead of sample variance. Note that sum of payoffs and sum of squares are statistically equivalent to sample means and variance.

We can encounter some difficulties in specifying the strategy sets and/or sets of types if these are infinite. To deal with these, we make an assumption that an interval can be specified for each attribute (thus, also constraining strategies to be real numbers). If we are dealing with a game of incomplete information, we may also need to use the *directions* interface to specify the distribution of types for each player. In the case when the set of types is finite, this is straighforward, since we would simply specify the probability of each type being played. When there are infinitely many types, we will make the assumption that the distribution of types is piecewise uniform, and that each attribute of the type vector is independently distributed. Then we can specify a value of the type density for each interval.[9]

6.2 Output Options

As indicated in Figure 2, all output will be in the form of a set of solutions. Internally, it will be represented by an XML tree. The XML tree will be passed to a formatter specified through the *directions* interface which will then present the data in some usable form to the world (e.g., to an analyst). For example, the user of EGAT may want the solutions in a textfile, and will therefore designate a "text formatter" to be used.

Since we cannot predict every possible use that the toolkit can have, we allow for considerable generality in output. In order to achieve such generality, we build the XML tree of solutions based on an XML schema specified in directions, against which the XML tree can be validated. Thus, this toolkit can in principle support any type of solution, as long as the appropriate XML schema and at least one corresponding output formatter is provided.[10]

The generality that will thus be built into the toolkit has its drawback: it makes user friendliness and especially developer friendliness an elusive goal. This is indeed a common conflict between manageability and extensibility, and we fully recognize that there will be most common uses of EGAT which will have to be provided with greater usability. Thus, an important part of a toolkit would be to create several "common use" XML Schema formatters, particularly those that support all the analysis tools that will actually be provided. Then, it will be up to the particular developers of analysis modules to resolve the issue of output format, which will be somewhat transparent to the end user. Overall, I feel that this design will result in considerable modularity and will consequently be beneficial in the long-term development of the toolkit.

I envision that a typical user will use the toolkit to determine or approximate the set of (approximate) Nash equilibrium outcomes and payoffs of a particular strategic interaction. Of course, as I have mentioned, if a user is a designer analyzing a particular strategic scenario, he may also need to know the probability that a particular strategic assessment (in terms of actual Nash or ϵ-Nash equilibria) is accurate. I plan to provide basic input and output interfaces to support such operations as a fundamental part of the toolkit.

Even if a designer is only concerned about Nash equilibrium outcomes, in some cases analysis may be restricted to a sample equilibrium, while in others the designer needs to know or approximate all Nash equilibria. Furthermore, the designer may be concerned about agents that are bounded-rational, and may have a model of play that incorporates certain aspects of rationality. For example, the designer may have a probabilistic belief that suggests that strategies that are undominated are

[9] When the distribution over types is not specified, it could be inferred from data using, for example, density estimation methods [Hastie et al., 2001].

[10] Naturally, we presume that there is also an analyzer that actually produces the particular solutions, or else the point is entirely moot.

considerably more likely to be played than dominated strategies. Consequently, the output will have to include sets of strategy profiles of players, and, perhaps, a probability distribution that profiles in some specified set are Nash equilibria, undominated, dominant, or have other strategically relevant properties. An entirely different output format would need to be provided to present an anlyst with payoff functions (usually approximate) of players, or an approximation of $\epsilon(r)$, although both of these would certainly be reasonable things to expect the toolkit to provide. All of these can be provided as solutions by specifying an appropriate solution format, as the internal XML tree is in practice agnostic to the format of data it will contain.

7 Finite Empirical Game Analysis Methods

For any reasonably small finite game we can usually find a set of solutions—Nash equilibria, for example—numerically. Similarly, given an empirical game as defined in Section 2.3, we can estimate or approximate a set of solutions of the actual (underlying) game.

I am aware of a number of research efforts that concentrate on formulating a game based on payoff function simulations in terms of a normal-form game that is subsequently solved using a standard finite-game solver. Reeves et al. [2005] analyzed the Market-Based Scheduling game using replicator dynamics as a large-game solution tool. In order to improve efficiency of estimates, the authors interleaved sampling and solving, drawing preferences according to Nash equilibrium population proportions based on the previous set of samples. Walsh et al. [2002] demonstrated analysis methods for finite games with small numbers of strategies, also using replicator dynamics. They incorporated rudimentary sensitivity anlaysis and used the size of the basin of attraction of replicator dynamics solutions to assess the likelihood of each of the game equilibria. The idea of thus deriving distributions of play has implications on mechanism design, as the designer can use the resulting distribution in his definition of $W(\mathcal{N}(\theta), \theta)$, an idea that has already been explored at a rudimentary level by Phelps et al. [2004].

In past work, I collaborated with several co-authors to present several methods for analyzing small finite games [Wellman et al., 2005a], applying them to analyze TAC/SCM 2003 strategic initial procurement of supplier components by the manufacturing agents. The crux of this work was comprised of data collection to obtain a description of three empirical games: one with six symmetric agents and two strategies, another with five symmetric agents and two strategies, and the last with six symmetric agents and three strategies. Nash equilibrium analysis of the first two games, as well as basic statistical analysis of equilibria, supported the conclusion that there was significant difference in terms of producer welfare between the first two games above. I also used a form of control variates to reduce variance of the mean estimates [L'Ecuyer, 1994, Ross, 2001].

In all of the research efforts described above, the set of Nash equilibria of the game which is obtained using sample average payoffs for each strategy profile is used as an *estimator* of the actual set of Nash equilibria of the game. This seems to be a sensible approach, and in Vorobeychik et al. [2005a] I provide some convergence results of such estimators. Treating these solutions as estimators suggests also that we can assess our confidence in them based on particular assumptions about distribution of our payoff samples. In what follows, I describe my approach to assessing the quality of Nash equilibrium estimation, as well as estimation of several other solution concepts, obtained under different assumptions about sampling noise.

7.1 Distribution-Free Bounds

The first lemma provides a (deterministic) bound on $\epsilon(r)$, where r is some strategy profile in the set of joint strategies, given that we have a bound on the quality of the payoff function approximation for every point in the domain.

Lemma 1. *Let $u_i(r)$ be the underlying set of payoff functions for all players and $\hat{u}_i(r)$ be an approximation of $u_i(r)$ for each $i \in I$. Suppose that $|u_i(r) - \hat{u}_i(r)| \leq \delta$ for strategy profiles r and $\forall (a_i, r_{-i}) : a_i \in A_i$. Then $|\epsilon(r) - \hat{\epsilon}(r)| \leq 2\delta$.*

The proof of this Lemma as well as all other non-trivial proofs are presented in the Appendix.

Now I will derive a distribution-free bound, first for pure strategy and thereafter for mixed strategy approximate equilibria. For pure strategy bounds, I assume that the following condition holds pointwise on R:

$$\Pr\{|u(r) - \hat{u}(r)| \geq \epsilon\} \leq \delta. \tag{2}$$

The desired probabilistic bound is then described by the following lemma.

Proposition 2. *Suppose (2) holds* pointwise *on R. Then*

$$\Pr\{|\epsilon(r) - \hat{\epsilon}(r)| \geq 2\gamma\} \leq (K+1)m\delta = 1 - \alpha,$$

where $K = \max_{i \in I} |A_i|$.

Application of Chebyshev's inequality and Chernoff bound yields the following corollaries for $a \in A$:

Corollary 3. *For any random variable with finite variance,*

$$\Pr\{|\epsilon(a) - \hat{\epsilon}(a)| \geq \gamma\} \leq \frac{4(K+1)m\sigma^2}{n\gamma^2}.$$

Corollary 4. *For a random variable that is bounded between L and U,*

$$\Pr\{|\epsilon(a) - \hat{\epsilon}(a)| \geq \gamma\} \leq m(K+1)\exp\left\{-\frac{\gamma^2 n}{2(U-L)^2}\right\}.$$

To find a distribution-free bound for a mixed strategy profile, we need a somewhat stronger assumption:

Lemma 5. *Suppose (2) holds* uniformly *on A. Then*

$$\Pr\{|u(s) - \hat{u}(s)| \geq \epsilon\} \leq \delta \ \forall s \in S.$$

Theorem 6. *Suppose (2) holds* uniformly *on A and let $s \in S$. Then*

$$\Pr\{|\epsilon(s) - \hat{\epsilon}(s)| \geq 2\epsilon\} \leq m(K+1)\delta.$$

Proof. By Lemmas 2 and 5. □

Corollary 7. *For any random variable with finite variance, $s \in S$,*

$$\Pr\{|\epsilon(s) - \hat{\epsilon}(s)| \geq \gamma\} \leq \frac{4|A|(K+1)m\sigma^2}{n\gamma^2}.$$

Corollary 8. *For a random variable that is bounded between L and U, $s \in S$,*

$$\Pr\{|\epsilon(s) - \hat{\epsilon}(s)| \geq \gamma\} \leq m|A|(K+1)\exp\left\{-\frac{\gamma^2 n}{2(U-L)^2}\right\}.$$

For general applicability of the above result, I observe in the following simple proposition that we can use them to bound a probability that a particular profile is a δ-Nash equilibrium for a given δ.

Proposition 9. *Let δ be given. Then,* $\Pr\{\epsilon(r) \leq \delta\} \geq Pr\{|\epsilon(r) - \hat{\epsilon}(r)| \leq \delta - \hat{\epsilon}(r)\}$.

Proof.

$$\Pr\{\epsilon(r) \leq \delta\} = \Pr\{\epsilon(r) - \hat{\epsilon}(r) \leq \delta - \hat{\epsilon}(r)\} \geq \Pr\{|\epsilon(r) - \hat{\epsilon}(r)| \leq \delta - \hat{\epsilon}(r)\}.$$

□

7.2 Confidence Bounds for Finite Games with Normal Noise

First, I explore games in which all agents have finite (and small) pure strategy sets, A. Thus, it is feasible to sample the entire payoff matrix of the game. To derive a generic probabilistic bound for a profile $r \in R$, recall that in the Bayesian framework, $u_i(\cdot)|\bar{u}_i(\cdot)$ are random variables. Then, if $u_i(\cdot)|\bar{u}_i(\cdot)$ are independent for all i, we have the following result (in the sequel I omit conditioning on $\bar{u}_i(\cdot)$ for brevity):

Proposition 10.

$$\Pr\left(\max_{i \in I}\max_{b \in R_i} u_i(b, r_{-i}) - u_i(r) \leq \epsilon\right) = \prod_{i \in I}\int_{\mathbb{R}}\prod_{b \in R_i \setminus r_i} \Pr(u_i(b, r_{-i}) \leq u + \epsilon) f_{u_i(r)}(u)du,$$

where $f_{u_i(a)}(u)$ is the pdf of $N(\bar{u}_i(a), \sigma_i(a))$.

7.2.1 Bounds on Pure Strategy Profiles

In this section I will derive a probabilistic bound on $\epsilon(a)$, where $a \in A$, given the assumption of normal, additive noise and an improper prior on the actual payoffs.[11] The posterior distribution of the optimum mean of n samples was derived in [Chang and Huang, 2000]:

$$\Pr(u_i(a) \leq c) = 1 - \Phi\left[\frac{\sqrt{n_i(a)}(\bar{u}_i(a) - c)}{\sigma_i(a)}\right], \tag{3}$$

where $a \in A$ and $\Phi(\cdot)$ is $N(0,1)$ distribution function. Applying Lemma 10, we get

$$\Pr\left(\max_{i \in I}\max_{b \in A_i \setminus a_i} u_i(b, a_{-i}) - u_i(a) \leq \epsilon\right) = \prod_{i \in I}\int_{\mathbb{R}}\prod_{b \in A_i \setminus a_i} \Pr(u_i(b, a_{-i}) \leq u + \epsilon) f_{u_i(a)}(u)du = 1 - \alpha. \tag{4}$$

[11] These bounds are also presented in [Vorobeychik et al., 2005a].

7.2.2 Bounds on Mixed Strategy Profiles

Proposition 11. *Let $s \in S$ be a mixed strategy profile. Then,*

$$\Pr\left(\max_{i \in I} \max_{b \in A_i} u_i(b, s_{-i}) - u_i(s) \leq \epsilon\right) = \prod_{i \in I} \int_{\mathbb{R}} \prod_{b \in A_i} \left[\Pr(W_i(b) \leq u + \epsilon)\right] f_{W_i^*}(u) du,$$

where

$$\Pr(W_i(b) \leq u + \epsilon) = 1 - \Phi\left[\frac{\sum_{c \in A_{-i}} \bar{u}_i(b, c) s_{-i}(c) - u - \epsilon}{\sqrt{\sum_{c \in A_{-i}} \frac{\sigma_i^2(b,c) s_{-i}^2(c)}{n_i(b,c)}}}\right]$$

and

$$W_i^* \sim N\left(\sum_{a \in A} \bar{u}_i(a) s(a), \sum_{a \in A} \frac{\sigma^2(a) s^2(a)}{n_i(a)}\right).$$

7.3 Alternatives to Nash Equilibria

While the Nash equilibrium has long been the primary, and, arguably, most compelling solution concept for games, many have expressed dissatisfaction with it in the context of real mechanism design problems and agent strategic considerations. For example, Erev and Roth [1998] provide experimental evidence that a reinforcement learning algorithm tends to be a better predictor of actual play in games with a unique equilibrium. In a similar vein, Selten [1991] presents a series of arguments against the Bayesian rationality as a reasonable predictor and, even, as an effective normative system. One appealing alternative that has been proposed is Rules of Thumb [Rosenthal, 1993a,b], that is, strategies that are essentially simple rules or series of rules conditioned on context that have proven effective over time. A similar notion of heuristic strategies has been studied by Walsh et al. [2002], and there has been work to find a Nash equilibrium in a relatively small set of heuristic strategies [Walsh et al., 2002, Reeves et al., 2005].

In this section, I describe several concepts that allow us to consider strategic choices of agents that are, while not necessarily Nash equilibria, still have appealing properties that may qualify them as reasonable *rules of thumb*. Thus, I expand the set of solutions that a designer may consider as plausible agent play, and thereby allow the mechanism designer to make the most appropriate choice for a particular problem.

7.3.1 Nearly Dominant Profiles

Dominant strategies are, perhaps, the most compelling solution concept in games. Unfortunately, aside from the common textbook examples, such as the famed Prisoner's Dilemma, they rarely exist in real strategic scenarios. In comparison, a Nash equilibrium is guaranteed to exist in all finite games. An intuitive property of a rule of thumb is that it *usually* works *reasonably* well. The way we can translate this idea into a solution concept is by introducing an ϵ-dominant strategy. While dominant strategies are rare, strategies that are nearly dominant may be more common, and may indeed provide a solid basis for certain rules of thumb. Of course, there will always be an ϵ-dominant strategy if we set ϵ to be high enough (as long as the payoff functions are bounded). However, once ϵ is large, such strategies are no longer nearly dominant in any meaningful way. Still, this solution concept may be a useful and reasonably compelling way to model agent play without appealing to the hyperrational Nash equilibrium. From a players' viewpoint, nearly dominant pure strategies are easy to find with respect to a given game, although the algorithmic question of finding the entire set of (pure and mixed) nearly dominant profiles may be a bit more involved and will be subject of future work. From a designer's viewpoint, they are reasonable things to expect agents to play. Below, I provide a formal definition and probabilistic bound for nearly-dominant profiles (that is, profiles in which each player's strategy is nearly dominant).

Definition 3. *A profile, r is ϵ-dominant if* $\forall i \in I, \forall t \in R_{-i}$,

$$u_i(r_i, t) + \epsilon \geq u_i(r', t),\ \forall r' \in R_i.$$

Proposition 12.

$$\Pr\left(r \text{ is } \epsilon\text{−dominant}\right) = \prod_{i \in I} \prod_{t \in R_{-i}} \int_{\mathbb{R}} \prod_{r' \in R_i \setminus r_i} \Pr\{u_i(r', t) \leq u + \epsilon\} f_{u_i(r_i,t)}(u) du.$$

7.3.2 Nearly Undominated Profiles

In his seminal work, Pearce [1984] describes the notion of rationalizable strategies. While the set of all rationalizable strategies is not always identical to the set of strictly undominated strategies, the two concepts are closely related, and are indeed appealing on similar grounds. The argument of Pearce was that the Nash equilibrium concept was too strong to describe actual behavior. By weakening it to a set of *plausible* strategy profiles that may be observed, actual behavior may be explained, although no longer modeled precisely.

While the idea that players are unlikely to play a profile that is strictly dominated (or not rationalizable) is very intuitive, there is experimental evidence to suggest that dominated strategies (for example, cooperative play in Prisoner's Dilemma) may indeed be played in practice [Cooper and Ross, 1996]. As a consequence, I introduce here an even weaker concept of nearly undominated or ϵ-undominated strategies, which include strategies that, while dominated, are very close to being optimal for some strategy profile that other agents may play. This property may keep such strategies as plausible rules of thumb. Below, I give a definition and probabilistic bounds for nearly-undominated strategy profiles.

Definition 4. *A profile, r, is ϵ-undominated if* $\forall i \in I, \exists t \in R_{-i}$, *such that*

$$u_i(r_i, t) + \epsilon \geq u_i(r', t),\ \forall r' \in R_i.$$

Proposition 13.

$$\Pr\left(r \text{ is } \epsilon\text{−undominated}\right) = \prod_{i \in I} \left[1 - \prod_{t \in R_{-i}} \int_{\mathbb{R}} \left(1 - \prod_{r' \in R_i \setminus r_i} \Pr\{u_i(r', t) \leq u + \epsilon\}\right) f_{u_i(r_i,t)}(u) du\right].$$

7.3.3 Safety of Pure and Mixed Profiles

As a part of sensitivity analysis of an empirical equilibrium, it may be useful to consider sensitivity not just to noise in the empirical game, but also to the deviations by other agents. To that end, I define the notion of δ-safety.

Definition 5. *Let R_{-i} be the joint space of deviations of agents other than i. A profile r is δ_i-safe for agent i if*

$$\delta_i(r) \geq \max_{t \in R_{-i}} (u_i(r) - u_i(r_i, t)).$$

A profile s is then δ-safe if it is δ_i-safe for all agents, that is, if

$$\delta(r) \geq \max_{i \in I} \max_{t \in R_{-i}} (u_i(r) - u_i(r_i, t)).$$

Alternatively, r is δ-safe if, for every player $i \in I$,

$$u_i(r) \leq u_i(r_i, t) + \delta, \ \forall t \in R_{-i}$$

A Proposition analogous to Proposition 10 gives a general probabilistic bound on $\delta(r)$:

Proposition 14.

$$\Pr\left(\max_{i \in I} \max_{t \in R_{-i}} (u_i(r) - u_i(r_i, t)) \leq \delta\right) = \prod_{i \in I} \int_{\mathbb{R}} \prod_{t \in R_{-i}} \left[1 - \Pr(u_i(r_i, t) \leq u - \delta)\right] f_{u_i(r)}(u) du.$$

The proof is analogous to cases already discussed, with r a pure or a mixed profile.

I do not see the notion of δ-safety as having much independent value. Instead, I view it as a useful way to distinguish particular types of rule-of-thumb strategies that players may consider. For example, we can imagine that in a set of approximate Nash equilibria, there may be profiles that would be extremely sensitive to deviations by players, and, therefore, have a higher bound on δ-safety. The notion of δ-safety thus provides a mechanism designer with an additional assessment of likelihood of play by indirectly accounting for risk aversion of agents without having to quantify it.

Given this motivation, we need a way to answer questions about probability of a particular profile being a δ-safe ϵ-equilibrium for a given δ and ϵ, as well as similar questions about other solution notions I have developed. I present these below.

First, I present a bound on probability that a profile r is a δ-safe ϵ-Nash equilibrium.

Proposition 15.

$$\begin{aligned} &\Pr(r \text{ is } \delta\text{−safe}, \ \epsilon\text{−Nash}) = \\ &\quad \Pr([\max_{i \in I} \max_{t \in R_{-i}} (u_i(r) - u_i(r_i, t)) \leq \delta] \& [\max_{i \in I} \max_{r' \in R_i} (u_i(r', r_{-i}) - u_i(r)) \leq \epsilon]) = \\ &\quad \prod_{i \in I} \int_{\mathbb{R}} \left[\left(\prod_{t \in R_1} \Pr\{u_i(r_i, t) \geq u - \delta\}\right) \left(\prod_{r' \in R_i \setminus r_i} \Pr\{u_i(r', r_{-i}) \leq u + \epsilon\}\right)\right] f_{u_i(r)}(u) du. \end{aligned}$$

We can combine δ-safety and ϵ-dominance or ϵ-dominatedness analogously, although in these cases the derivations are somewhat more laborious.

Proposition 16.

$$\begin{aligned} &\Pr\{r \text{ is } \delta\text{−safe}, \ \epsilon\text{−dominant}\} = \\ &\quad \Pr\{\max_{i \in I} \max_{t \in R_{-i}} [u_i(r) - u_i(r_i, t)] \leq \delta \ \& \max_{i \in I} \max_{t \in R_{-i}} \max_{r' \in R_i} [u_i(r', t) - u_i(r_i, t)] \leq \epsilon\} \geq \\ &\quad \prod_{i \in I} \int_{\mathbb{R}} \left[\left(\prod_{t \in R_{-i}} \Pr\{u_i(r_i, t) \geq u - \delta\}\right) \left(\prod_{r' \in R_i \setminus r_i} \Pr\{u_i(r', t) \leq u + \epsilon - \delta\}\right)\right] f_{u_i(r)} du. \end{aligned}$$

Proposition 17.

$$\Pr\{r \text{ is } \delta\text{−safe, } \epsilon\text{−undominated}\} =$$

$$\Pr\{\max_{i\in I}\max_{t\in R_{-i}\backslash r_{-i}}[u_i(r)-u_i(r_i,t)]\le\delta \;\&\max_{i\in I}\min_{t\in R_{-i}}\max_{r'\in R_i\backslash r_i}[u_i(r',t)-u_i(r_i,t)]\le\epsilon\} =$$

$$\prod_{i\in I}\int_{\mathbb{R}}\left[\prod_{t\in\mathbb{R}_{-i}\backslash r_{-i}}\Pr\{u_i(r_i,t)\ge u-\delta\}\right]\left[1-\left(1-\prod_{r'\in R_i\backslash r_i}\Pr\{u_i(r',r_{-i})\le\epsilon+u\}\right)\times\right.$$

$$\left.\times\left(\prod_{t\in\mathbb{R}_{-i}\backslash r_{-i}}\frac{1}{\int_{u-\delta}^{\infty}f_{u_i(r_i,t)}(v)dv}\int_{u-\delta}^{\infty}\{1-\prod_{r'\in R_i\backslash r_i}\Pr(u_i(r',t)\le\epsilon+v)\}f_{u_i(r_i,t)}(v)dv\right)\right]f_{u_i(r)}(u)du.$$

8 Infinite Game Analysis

As we have seen in the immediately preceding sections, finite games already possess considerable difficulties for empirical game analysis when payoffs are only available via noisy samples. Imagine now that the game has many players and many (perhaps infinite) number of strategies for each player. The difficulties that arise from noisy simulations are almost overshadowed by the sheer impossibility of obtaining even a single sample for every strategy profile that could be played. Furthermore, even if it was indeed possible to specify a payoff function of the game, no finite game solver would be able to find even a single Nash equilibrium in all but very simple cases.

In the following sections, I will primarily consider the case of infinite games, recognizing that when a finite game is too large, the same methods will need to be applied to it as well. Frequently in my discussion I will make use of a convenient assumption that the game is also symmetric, as in definition 2. In most cases, this assumption is not restrictive and is made primarily for clarity of exposition.

Any analysis of infinite games that relies only on a finite sample from the payoff functions must naturally make assumptions that the game possesses some structure making it amenable to generalization based on limited information. While we can never fully ascertain that any assumption about structure actually holds in a game without prior knowledge, generally I expect that games of interest will indeed exhibit enough smoothness to justify my approaches.

8.1 Finite Game Approximations

Perhaps the most natural way to tackle an infinite game is via a finite approximation. This can be done as follows. First, let us say that we already possess a data set of strategy profiles and corresponding samples from the payoff function. For simplicity, let us even say that we have obtained the sample average payoff for each strategy profile that exists in our data.

Let $\hat{A}_i$ be the set of strategies of player i such that for every $a_i \in \hat{A}_i$, there is a profile b in the data set which includes a_i. In words, $\hat{A}_i$ is the set of strategies of player i that appears in the dataset. If, furthermore, for each player i and for each b that includes $a_i \in \hat{A}_i$, the data set includes all the profiles (a'_i, b_{-i}), with $a'_i \in \hat{A}_i$, let us say that our data set constitutes a *clique*. In such a case, we can use a generic finite-game solver to obtain a Nash equilibrium (or some other solution) for this *restricted* game.

Imagine that we are now given a data set which does constitute a *clique*. As an extension of the above approach, we can find a *maximal clique* and use a Nash equilibrium computed on the resulting restricted game as an estimate of a Nash equilibrium of the actual game. Unfortunately, the problem of computing the maximal clique is known to be NP-Complete [Garey and Johnson, 1979].[12]

Another downside of restricting Nash equilibrium estimation to a subset of the data that constitutes a maximal clique is that we are in effect throwing information away, and when we have worked so hard to obtain it, this appears to be a tremendous waste. An alternative that does use all the profiles in Nash equilibrium estimation is one that uses a set of profiles with ϵ-bounds below some fixed threshold as an estimate of a set of Nash equilibria. I have more to say about this method below.

As I had mentioned earlier, there are various questions we may like to ask about games based on a data set of payoffs. A restriction of infinite games to finite games provides a way to estimate an answer to any such question. For example, a set of Nash equilibria of a finite restricted game would be our estimate of a set of Nash equilibria of the underlying infinite game.[13] Similarly, we can use the finite approximations to derive probabilistic bounds on the proximity (in $\epsilon(r)$ pseudometric) of a profile r to equilibrium, as I describe in the following section.

8.2 Confidence Bounds Using Finite Game Approximations

Suppose that we are trying to estimate a Nash equilibrium for a game $\Gamma = [I, R, u(\cdot)]$ with R infinite. Let $R_k \subset R$ be finite and define $\Gamma_k = [I, R_k, u(\cdot)]$ to be a finite approximation of Γ. To draw any conclusions about Γ based on its finite approximation we must make some assumptions about the structure of the actual payoff functions on the infinite domain. I assume that the payoff functions, $u_i(\cdot)$ of all players satisfy the Lipschitz condition with Lipschitz coefficient B.

Define d to be the maximum distance from a point in $R_{k,i}$ to its closest neighbor in R_i:

$$d = \max_{i \in I} \sup_{r \in R_i} \inf_{r' \in R_{k,i}} \{|r - r'|\} < \infty.$$

Then if r is an ϵ-Nash of Γ_k with probability $1-\alpha$, then it is an $(\epsilon + Bd)$-Nash of Γ with probability at least $1 - \alpha$. Consequently, we have the following bound:

Proposition 18.

$$\Pr\left(\max_{i \in I} \sup_{t \in R_i} u_i(t, r_{-i}) - u_i(r) \leq \epsilon\right) \geq \prod_{i \in I} \int_{\mathbb{R}} \prod_{t \in R_{k,i}} \Pr(u_i(t, r_{-i}) \leq u + \epsilon - Bd) f_{u_i(r)}(u) du.$$

8.3 Learning Payoff Functions

An important consequence of restricting the sets of strategies to finite subsets is that we are also thereby restricting our solutions (e.g., Nash equilibrium estimates) to lie in the mixed strategy space of the resulting finite games. In [Vorobeychik et al., 2005b] several coauthors and I proposed an alternative. The data set of strategy profiles and corresponding payoffs, which we envision to be in the form $(a, u(a))$ (I will assume for simplicity that the game is symmetric and $u(a) = u_1(a)$ is the symmetric payoff function), can serve as the data set for a supervised learning problem. Given such

[12] My definition of a clique can be mapped to a graph by making each profile a node in a graph and defining an edge to be between nodes corresponding to profiles that are different in exactly one strategic choice.

[13] When the payoffs in the data set are available through noiseless samples, a more appropriate term would be *approximation*, rather than estimate.

a framework, we can select a learning model—for example, polynomial regression, neural network, or SVM [Hastie et al., 2001]—and fit that model to our data set, with a providing the input and $u(a)$ the output.

While an approximate game $[I, R, \hat{u}(r)]$, where $\hat{u}(r)$ is the output of learning, can in principle be used to get an estimate (or approximation) to any solution of the underlying game $[I, R, u(r)]$ desired, in practice, of course, no numerical tool exists to do this. In special cases, such as polynomial regression, we can indeed use numerical tools to find Nash equilibria of the learned game. For most models that we will use, however, we must use some finite approximation of the learned game in order to obtain an approximate solution.

A natural question that arises is how the different supervised learning techniques compare to each other as well as to finite approximation for various solution types that would be desired by an empirical analyst. In [Vorobeychik et al., 2005b] we present numerous empirical results that compare learning and finite game approximation methods based on the quality of Nash equilibrium estimation (approximation) in the $\epsilon(r)$ pseudometric.

8.4 Using $\epsilon(s)$ as target

While the data in the form $(a, u(a))$ is the most natural target of learning if we have an empirical game, there is another alternative, which was explored by Vorobeychik et al. [2005b]: transforming the data set $(a, u(a))$ into $(a, \epsilon(a))$. Subsequently, we can use $\epsilon(a)$ as the target of learning. Nash equilibrium would then be computed as the global minimum of the output function of learning, $\hat{\epsilon}(a)$, with the function value of 0. Experimental results suggest that in some cases it does produce better Nash equilibrium estimates than the learning methods that use $u(a)$ as target [Vorobeychik et al., 2005b].

8.5 Learning Bayes-Nash Equilibria

Let us suppose that we have an extended data set of experience, (a, v, t), where t is the vector of player types that play the pure strategy profile a and accrue the resulting payoffs, v. Given such a data set, we can extend the input to our function approximation models to include the players' type vector, t as the additional set of inputs. A particularly appealing model class is one studied by Reeves and Wellman [2004]. For any model in this class Reeves and Wellman derived a method for computing a best response to a piecewise-linear strategy function (of players' types) exactly. This best-response finder can also be used in an iterative best-response dynamic to find Bayes-Nash equilibria.

If we know the actual distribution of players' types, we can use it directly in an equilibrium computation tool, whereas an unknown type distribution can be estimated from the data using density estimation methods [Hastie et al., 2001].

9 Simulation Control and Active Learning

Simulation control in the context of games has yet to be extensively studied. Walsh et al. [2004] use an information-theoretic method for simulation control, while Reeves et al. [2005] interleave replicator dynamics and simulation control, but these two reports appear to be the extent to which this problem has been explored in the literature.

The fundamental question here is to select the best strategy profile to sample next given a set of profiles already samples. The question thus phrased is woefully incomplete, as it offers no meaning of "best". In both of the related papers cited here, the target of analysis is Nash equilibrium estimation. In this setting, the "best" profile would result in the best estimate or approximation of a Nash equilibrium in expectation. Walsh et al. [2004] experimentally evaluated their information-theoretic methods to show that the resulting Nash equilibrium approximations were closer to Nash equilibrium in the ϵ metric.[14] Reeves et al. [2005] interleave replicator dynamics with sample selection based on steady-state population proportions of each strategy, but do not evaluate its efficacy.

Neither of the methods I have cited attempts to tackle the problem of simulation control directly by trying to minimize the expected error of the estimate of Nash equilibrium. The way I would suggest formulating this problem is to take the measure of error to be mean squared deviation in the relevant metric (which I will usually take to be the ϵ-metric). Another alternative tactic for simulation control is to use probabilistic bounds I have developed above. The resulting method would select the next profile to be sampled with probability that is proportional to the probability that this profile is a Nash equilibrium. I expect to try both of these ideas in the future. In the meantime, I present several additional methods which I either have already tried or which I can describe more concretely.

9.1 Equilibrium Search in Strategy Space

When we have access to a simulator and we are interested in estimating the set of Nash equilibria, we can use directed search through profile space to estimate the set of Nash equilibria, which I describe here after presenting some additional notation.

Definition 6. *A* strategic neighbor *of a pure strategy profile a is a profile that is identical to a in all but one strategy. Define $S_{nb}(a, \mathcal{D})$ as the set of all strategic neighbors of a available in the data set $\mathcal{D}$. Similarly, define $S_{nb}(a, \tilde{\mathcal{D}})$ to be all strategic neighbors of a not in $\mathcal{D}$. Finally, for any $a' \in S_{nb}(a, \mathcal{D})$ define the deviating agent as $i(a, a')$.*

Definition 7. *The ϵ-bound, $\hat{\epsilon}$, of a pure strategy profile a is defined as $\max_{a' \in S_{nb}(a,\mathcal{D})} \max\{u_{i(a,a')}(a') - u_{i(a,a')}(a), 0\}$. We say that a is a* candidate δ-equilibrium *for $\delta \geq \hat{\epsilon}$.*

When $S_{nb}(a, \tilde{\mathcal{D}}) = \emptyset$ (i.e., all strategic neighbors are represented in the data), a is *confirmed* as an $\hat{\epsilon}$-Nash equilibrium.

My search method operates by exploring deviations from promising candidate equilibria. I refer to it as "BestFirstSearch", as it selects with probability one a strategy profile $a' \in S_{nb}(a, \tilde{\mathcal{D}})$ that has the smallest $\hat{\epsilon}$ in $\mathcal{D}$.

Finally I define an estimator for a set of Nash equilibria.

Definition 8. *For a set K, define $Co(K)$ to be the convex hull of K. Let B_δ be the set of candidates at level δ. Define $\hat{\phi}^*(\theta) = Co(\{\phi(a) : a \in B_\delta\})$ for a fixed δ to be an estimator of $\phi^*(\theta)$.*

In words, the estimate of a set of equilibrium outcomes is the convex hull of all aggregated strategy profiles with ϵ-bound below some fixed δ. This definition allows us to exploit structure arising from the aggregation function. If two profiles are close in terms of aggregation values, they

[14] The ϵ metric uses $\epsilon(r)$ to evaluate the quality of Nash equilibrium approximation by a profile r.

may be likely to have similar ϵ-bounds. In particular, if one is an equilibrium, the other may be as well. I present some theoretical support for this method of estimating the set of Nash equilibria below. I used this method in [Vorobeychik et al., 2005a] in order to estimate the Nash equilibrium correspondence.

9.2 Active Learning

My treatment of empirical game estimation using supervised learning has thus far only considered a static data set of examples that is available at the time of function approximation and assumed that no further data points can be generated thereafter. Alternatively, we can envision a simulator that produces data points on-demand. In such a setting, function approximation is often enhanced when we can select data points to sample in order to minimize the variance of the approximate function (thereby also reducing the mean squared error, since the bias of the model space remains constant). A number of techniques had been developed that leverage the specific properties of the model class under consideration in order to selectively sample the function domain [Cohn et al., 1994]. These fall under the rubric of *active learning*.

The primary goal of active learning is to improve the expected quality of function fit in terms of mean squared error. Our goal, however, is somewhat different. We do not necessary care whether the quality of function fit is improved, as long as we can extract a better approximation of a Nash equilibrium from the approximate payoff function. While the two goals are intimately related (a poor function fit will generally not represent the strategic interactions very well), the techniques for achieving each may well be different.

9.3 Active Learning with $\epsilon(a)$ as Target

In addition to using active learning to aid payoff function approximation, I would like to consider $\epsilon(a)$ as an alternative active learning target. Using $\epsilon(a)$ as a target may well provide considerable advantages in approximating Nash equilibria of a game, since $\epsilon(a)$ relates more directly to Nash equilibria, particularly if we use this function to measure the quality of our approximation!

9.4 Estimation of a Sample Nash Equilibrium Using Simulation Optimization

When the basic goal is to estimate just a sample Nash equilibrium solution to a game using noisy simulations, I would like to tap a related stream of research activity with a very active literature which addresses the problem of estimating an optimum of a function based on noisy simulations [Spall, 2003, Kiefer and Wolfowitz, 1952, Gurkan et al., 1994, Neddermeijer et al., 2000, Olafsson, 1999, Shi and Olafsson, 1997, Olafsson and Kim, 2001, Fleischer, 1995, Eglese, 1990, Alrefaei and Andradottir, 1995, Yakowitz et al., 2000, Moore et al., 1998, Anderson et al., 2000]. The motivation for this problem is similar to ours in that no closed-form description of the function to be optimized is available. Instead, a simulator is provided that gives an unbiased noisy sample of the objective function at a given query point. Numerous techniques have been developed, many with proven convergence properties, to estimate an optimum of the objective function.

Our setting is different from standard simulation optimization because we have an additional level of indirection between the information available through simulations and what we are trying to estimate. Specifically, the simulator provides us with unbiased samples of players' payoffs. Given such a simulator, we can use any simulation optimization method directly to estimate $\epsilon(a)$ for any strategy profile a, since that involves computing the optimum of each player's payoff function. However, we are interested in estimating Nash equilibria, which are *minima* of $\epsilon(a)$ over all $a \in A$ or, in the case of mixed strategy equilibria, the minima of $\epsilon(s)$ over the mixed strategy space S.

I would like to suggest a two-tiered simulation optimization framework to estimate pure strategy Nash equilibria of an empirical game. In the first (high-level) tier, I would use a simulation optimization method to minimize $\epsilon(a)$ over the pure strategy set. This, in turn, will rely on the second tier in which I estimate the best response of each player to strategy a, where a was selected for sampling in the first tier.

Clearly, even if our simulation optimization method of choice is consistent, we lose the assumption of unbiased samples in the first-tier level, since each estimate of $\epsilon(a)$ will typically not be unbiased. I hope that I can obtain convergence to actual Nash equilibria nevertheless. More significantly, I expect to observe experimentally whether such a two-tiered framework for estimating Nash equilibria is effective, particularly compared to other methods I propose.

10 Constructing Belief Distributions of Agent Play

The most straightforward way for a designer to use probabilistic bounds that assess how close a particular strategy profile is to a solution of choice is to incorporate them into the analysis of a particular game, and use them this way to make high-level decisions about designs that are extremely complex and not amendable to decision-theoretic approaches.

A more formally satisfying approach, however, would proceed to specify a probability distribution describing the likelihood of play of each pure strategy profile in the games induced by the design parameter settings, and select one that maximizes expected designer utility with respect to this distribution. It is noteworthy, however, that such decision-theoretic analysis leaves the actual process of forming believe distributions regarding agent play to the whim of the analyst. A theorist would say that in principle, the analyst would use prior information to determine his beliefs. Fine, I say, suppose that the analyst believes that Nash equilibrium strategy profiles are more likely to be played than other profiles. This still leaves ample ground for expressing the vague statement "more likely" as actual probabilities. And the formulation of such beliefs is further jeopardized when we are uncertain about which profiles are actually (approximate) Nash equilibria! Thus, I believe that in most applications an analyst needs principled methods for formulating beliefs, and in this section I propose several approaches for doing this.

The first approach I propose is to use the probabilistic bounds I have derived above in fully *decision-theoretic* mechanism design by defining the belief distribution of profiles that are played as follows. Suppose that a designer is relatively confident that agents will play profiles that belong to a particular solution class (e.g., Nash equilibrium profiles). Let's denote such a class by $\mathcal{C}$. Then, the designer could form the belief that "a profile a will be played" by

$$\Pr\{a \text{ is played}\} = \frac{\Pr\{a \in \mathcal{C}\}}{\sum_{a \in A} \Pr\{a \in \mathcal{C}\}}.$$

In words, the belief that a profile is played is proportional to the probability that this profile constitutes a solution in class $\mathcal{C}$.

For example, in the case when $\mathcal{C}$ is the set of Nash equilibria, the designer's belief distribution would be defined by

$$\Pr\{a \text{ is played}\} = \frac{\Pr\{\epsilon(a) \le 0\}}{\sum_{a \in A} \Pr\{\epsilon(a) \le 0\}}.$$

My purpose of using the abstract notion of "class" of solutions is that I could allow classes of solutions to include the notions of bounded rationality that I had discussed earlier. Thus, the designer may believe that agents will play an ϵ-Nash equilibrium only if it is δ-safe. A probability distribution over play may then be formed with respect to my results regarding $\Pr\{a \in \mathcal{C}_1 \& a \in \mathcal{C}_2\}$, where $\mathcal{C}_1$ is the set of ϵ-Nash equilibria and $\mathcal{C}_2$ is the set of δ-safe profiles.

While the heuristic distributions above are simple and convenient, a more compelling idea, perhaps, is for the designer to have a belief distribution over profiles in $\mathcal{C}$ *given that* $\mathcal{C}$ *is known*. For example, the designer may assume that agents will play a Nash equilibrium with uniform probability (i.e., play of each Nash equilibrium profile is equally likely from the designer's perspective). The designer, additionally, may have a belief over *games* that could be played by the agents given its parameter setting. The posterior distribution of agent play is then computed in a standard way:

$$\Pr\{a \text{ played}\} = E_{\text{games}}[\Pr\{a|\mathcal{C}\}].$$

Thus far, I have been unable to map this posterior distribution to any closed-form expression in terms of my probabilistic bounds. Nevertheless, it can be approximated numerically.

An interesting question now arises: how much does it matter whether the designer holds the first or the second type of belief? If the difference in results is not significant, then it makes much sense to employ the first, as it is quite simple and intuitive. If the difference is large, however, application of either will then be context-dependent. My research direction will thus evaluate expected values of the designer's utility function with respect to each type of belief, and compare the results. Particularly, I will compare the expected utilities at optima based on the different belief types. I hope to provide such comparisons for various types of designer's utility functions (for example, social utility or profit maximization from an auction).

References

Mahmoud H. Alrefaei and Sigrun Andradottir. A new search algorithm for discrete stochastic optimization. In *Winter Simulation Conference*, pages 236–241, 1995.

Brigham Anderson, Andrew Moore, and David Cohn. A nonparametric approach to noisy and costly optimization. In *International Conference on Machine Learning*, pages 17–24, 2000.

Ben Blume, 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, 2003.

Yi-Ping Chang and Wen-Tao Huang. Generalized confidence intervals for the largest value of some functions of parameters under normality. *Statistica Sinica*, 10:1369–1383, 2000.

Shih-Fen Cheng, Daniel M. Reeves, Yevgeniy Vorobeychik, and Michael P. Wellman. Notes on equilibria in symmetric games. In *AAMAS-04 Workshop on Game-Theoretic and Decision-Theoretic Agents*, New York, 2004.

David A. Cohn, Les Atlas, and Richard E. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

Russell Cooper and Thomas W. Ross. Cooperation without reputation: experimental evidence from prisoner's dilemma games. *Games and Economic Behavior*, 12:187–218, 1996.

R.W. Eglese. Simulated Annealing: A tool for operations research. *European Journal of Operations Research*, 46:271–281, 1990.

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

Mark Fleischer. Simulated Annealing: Past, present, and future. In *Winter Simulation Conference*, pages 155–161, 1995.

Daniel Friedman. Evolutionary games in economics. *Econometrica*, 59(3):637–666, May 1991.

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

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

M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.

Srihari Govindan and Robert Wilson. Structure theorems for game trees. *Proceedings of 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(1):65–86, 2003.

Gul Gurkan, A. Yonca Ozge, and Stephen M. Robinson. Sample-path optimization in simulation. In *Winter Simulation Conference*, pages 247–254, 1994.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.

P. Jean-Jacque Herings and Ronald Peeters. A globally convergent algorithm to compute all Nash equilibria for n-person games. *Annals of Operations Research*, 137:349–368, 2005.

Ehud Kalai and Ehud Lehrer. Rational learning leads to Nash equilibrium. *Econometrica*, 61(5): 1019–1045, September 1993.

J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23(3):462–466, September 1952.

Christopher Kiekintveld, Yevgeniy Vorobeychik, and Michael P. Wellman. An analysis of the 2004 supply chain management trading agent competition. In *IJCAI-05 Workshop on Trading Agent Design and Analysis*, 2005.

David M. Kreps. *Game Theory and Economic Modelling*. Oxford University Press, 1990.

Vijay Krishna. *Auction Theory*. Academic Press, 1st edition, 2002.

Pierre L'Ecuyer. Efficiency improvement and variance reduction. In *Winter Simulation Conference*, pages 122–132, 1994.

Andreu Mas-Colell, Michael D. Whinston, and Jerry R. Green. *Microeconomic Theory*. Oxford University Press, 1995.

Richard D. McKelvey and Andrew McLennan. Computation of equilibria in finite games. In *Handbook of Computational Economics*, volume 1, chapter 2, pages 87–142. 1996.

Richard D. McKelvey, Andrew M. McLennan, and Theodore L. Turocy. Gambit: Software tools for game theory, version 0.2005.06.13, 2005. URL `http://econweb.tamu.edu/gambit`.

John McMillan. Selling spectrum rights. *The Journal of Economic Perspectives*, 8(3):145–162, 1994.

Paul Milgrom. Putting auction theory to work: The simultaneous ascending auctions. *The Journal of Political Economy*, 108(2):245–272, April 2000.

Paul Milgrom and John Roberts. Adaptive and sophisticated learning in normal form games. *Games and Economic Behavior*, 3:82–100, 1991.

Andrew Moore, Jeff Schneider, Justin Boyan, and Mary Soon Lee. Q2: Memory-based active learning for optimizing noisy continuous functions. In *Proceedings of the Fifteenth International Conference of Machine Learning*, pages 386–394, 1998.

Roger B. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, February 1981.

John Nash. Non-cooperative games. *Annals of Mathematics*, 54:286–295, 1951.

H. Gonda Neddermeijer, Gerrit J. van Oortmarssen, Nanda Piersma, and Rommert Dekker. A framework for response surface methodology for simulation optimization. In *Winter Simulation Conference*, pages 129–136, 2000.

Sigurdur Olafsson. Iterative ranking-and-selection for large-scale optimization. In *Winter Simulation Conference*, pages 479–485, 1999.

Sigurdur Olafsson and Jumi Kim. Towards a framework for black-box simulation optimization. In *Winter Simulation Conference*, pages 300–306, 2001.

Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory*. The MIT Press, 1994.

David G. Pearce. Rationalizable strategic behavior and the problem of perfection. *Econometrica*, 52(4):1029–1050, July 1984.

Steve Phelps, Simon Parsons, and Peter McBurney. Automated agents versus virtual humans: an evolutionary game theoretic comparison of two double-auction market designs. In *Workshop on Agent Mediated Electronic Commerce VI*, 2004.

R.W. Porter, E. Nudelman, and Y. Shoham. Simple search methods for finding a Nash equilibrium. *Games and Economic Behavior*, 2006, to appear.

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, 2004.

Daniel M. Reeves, Michael P. Wellman, Jeffrey K. MacKie-Mason, and Anna Osepayshvili. Exploring bidding strategies for market-based scheduling. *Decision Support Systems*, 39:67–85, 2005.

J.G. Riley and W.F. Samuelson. Optimal auctions. *American Economic Review*, 71:381–392, 1981.

Robert W. Rosenthal. Rules of thumb in games. *Journal of Economic Behavior and Organization*, 22:1–13, 1993a.

Robert W. Rosenthal. Bargaining rules of thumb. *Journal of Economic Behavior and Organization*, 22:15–24, 1993b.

Sheldon M. Ross. *Simulation*. Academic Press, 3rd edition, 2001.

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

Alvin E. Roth and Elliott Peranson. The redesign of the matching market for american physicians: some engineering aspects of economic design. *The American Economic Review*, 89(4):748–780, September 1999.

Ariel Rubinstein. *Modeling Bounded Rationality*. The MIT Press, 1997.

Tuomas Sandholm, Andrew Gilpin, and Vincent Conitzer. Mixed-integer programming methods for finding Nash equilibria. In *20th National Conference on Artificial Intelligence*, pages 495–501, 2005.

Reinhard Selten. Evolution, learning, and economic behavior. *Games and Economic Behavior*, 3: 3–24, 1991.

Leyuan Shi and Sigurdur Olafsson. An integrated framework for deterministic and stochastic optimization. In *Winter Simulation Conference*, pages 358–365, 1997.

James C. Spall. *Introduction to Stochastic Search and Optimization*. John Wiley and Sons, Inc, 2003.

Yevgeniy Vorobeychik and Isaac Porche. Game theoretic methods for analysis of combat simulations. Ongoing RAND research, 2005.

Yevgeniy Vorobeychik, Christopher Kiekintveld, and Michael P. Wellman. Empirical mechanism design: Methods, with an application to a supply chain scenario. Technical report, University of Michigan, 2005a.

Yevgeniy Vorobeychik, Michael P. Wellman, and Satinder Singh. Learning payoff functions in infinite games. In *Nineteenth International Joint Conference on Artificial Intelligence*, pages 977–982, 2005b.

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*, 2002.

William E. Walsh, David Parkes, and Rajarshi Das. Choosing samples to compute heuristic-strategy Nash equilibrium. In *Agent-Mediated Electronic Commerce V*, 2004.

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

Michael P. Wellman, Daniel M. Reeves, Kevin M. Lochner, and Rahul Suri. Approximate strategic reasoning through hierarchical reduction of large symmetric games. In *Twentieth National Conference on Artificial Intelligence*, 2005b.

Sidney Yakowitz, Pierre L'Ecuyer, and Felisa Vazquez-Abad. Global stochastic optimization with low-dispersion point sets. *Operations Research*, 48(6):939–950, 2000.

Appendix

A Proofs

A.1 Proof of Lemma 1

First, we will need the following fact:

Claim. $|\max_{x\in X} f_1(x) - \max_{x\in X} f_2(x)| \le \max_{x\in X} |f_1(x) - f_2(x)|$, *where* X *is an arbitrary optimization domain and* $f_i(x)$ *an arbitrary function*

To show this, observe that

$$|\max_{x\in X} f_1(x) - \max_{x\in X} f_2(x)| = \begin{cases} \max_x f_1(x) - \max_x f_2(x) & \text{if } \max_x f_1(x) \ge \max_x f_2(x) \\ \max_x f_2(x) - \max_x f_1(x) & \text{if } \max_x f_2(x) \ge \max_x f_1(x) \end{cases}.$$

In the first case,

$$\max_{x\in X} f_1(x) - \max_{x\in X} f_2(x) \le \max_{x\in X}(f_1(x) - f_2(x)) \le \max_{x\in X} |f_1(x) - f_2(x)|.$$

Similarly, in the second case,

$$\max_{x\in X} f_2(x) - \max_{x\in X} f_1(x) \le \max_{x\in X}(f_2(x) - f_1(x)) \le \max_{x\in X} |f_2(x) - f_1(x)| = \max_{x\in X} |f_1(x) - f_2(x)|.$$

Consequently,

$$|\max_{x\in X} f_1(x) - \max_{x\in X} f_2(x)| \le \max_{x\in X} |f_1(x) - f_2(x)|.$$

Now, I use this to provide a bound on $|\epsilon(r) - \hat{\epsilon}(r)|$:

$$
\begin{aligned}
|\epsilon(r) - \hat{\epsilon}(r)| = |\max_{i \in I} \max_{a_i \in A_i} [\hat{u}(a_i, r_{-i}) - \hat{u}(r)] - \max_{i \in I} \max_{a_i \in A_i} [u_i(a_i, r_{-i}) - u_i(r)]| \leq \\
\max_{i \in I} \max_{a_i \in A_i} |[\hat{u}(a_i, r_{-i}) - \hat{u}(r)] - [u_i(a_i, r_{-i}) - u_i(r)]| = \\
\max_{i \in I} \max_{a_i \in A_i} |[\hat{u}(a_i, r_{-i}) - u_i(a_i, r_{-i})] + [u_i(r) - \hat{u}(r)]| \leq \\
\max_{i \in I} \max_{a_i \in A_i} |\hat{u}(a_i, r_{-i}) - u_i(a_i, r_{-i})| + |\hat{u}(r) - u_i(r)|.
\end{aligned}
$$

The result now follows from the assumption that $|u_i(r) - \hat{u}_i(r)| \leq \delta$ for strategy profiles r and $\forall (a_i, r_{-i}) : a_i \in A_i$.

A.2 Proof of Proposition 2

$$
\begin{aligned}
\Pr\{|\epsilon_n(r) - \epsilon(r)| \geq 2\gamma\} = \Pr\{|\max_{i \in I} \epsilon_{n,i}(r) - \max_{i \in I} \epsilon_i(r)| \geq 2\gamma\} \leq \\
\leq \Pr\{\max_{i \in I} |\epsilon_{n,i}(r) - \epsilon_i(r)| \geq 2\gamma\} \leq \\
\leq \Pr\{\bigcup_{i \in I} |\epsilon_{n,i}(r) - \epsilon_i(r)| \geq 2\gamma\} \leq \\
\leq \sum_{i \in I} \Pr\{|\epsilon_{n,i}(r) - \epsilon_i(r)| \geq 2\gamma\} \leq \\
\leq \sum_{i \in I} \Pr\{\bigcup_{q \in r \cup (A_i, r_{-i})} [|u_{n,i}(q) - u_i(q)| \geq \gamma]\} \leq \\
\leq \sum_{i \in I} (K+1)\delta \leq m(K+1)\delta.
\end{aligned}
$$

A.3 Proof of Lemma 5

$$
\begin{aligned}
&\Pr\{|u(s) - \hat{u}(s)| \geq \epsilon\} = \\
&= \Pr\{|u(s) - \hat{u}(s)| \geq \epsilon \,|\, |u(a) - \hat{u}(a)| \leq \epsilon \ \forall a \in A\} \times \Pr\{|u(a) - \hat{u}(a)| \leq \epsilon \ \forall a \in A\} + \\
&+ \Pr\{|u(s) - \hat{u}(s)| \geq \epsilon \,|\, \exists a \in A : |u(a) - \hat{u}(a)| > \epsilon\} \times \Pr\{\exists a \in A : |u(a) - \hat{u}(a)| > \epsilon\} \leq \delta.
\end{aligned}
$$

A.4 Proof of Proposition 10

$$
\begin{aligned}
\Pr\left(\max_{i \in I} \max_{b \in R_i \setminus r_i} u_i(b, r_{-i}) - u_i(r) \leq \epsilon\right) = \\
= \prod_{i \in I} \Pr(\max_{b \in R_i \setminus r_i} u_i(b, r_{-i}) - u_i(r) \leq \epsilon) = \\
= \prod_{i \in I} E_{u_i(r)} \Pr(\max_{b \in R_i \setminus r_i} u_i(b, r_{-i}) - u_i(r) \leq \epsilon | u_i(r)) = \\
= \prod_{i \in I} E_{u_i(r)} \left[\prod_{b \in R_i \setminus r_i} \Pr(u_i(b, r_{-i}) - u_i(r) \leq \epsilon | u_i(r))\right] = \\
= \prod_{i \in I} \int_{\mathbb{R}} \prod_{b \in R_i} \Pr(u_i(b, r_{-i}) \leq u + \epsilon) f_{u_i(r)}(u) du.
\end{aligned}
$$

A.5 Proof of Proposition 11

Define $W_i(b) = u_i(b, s_{-i})$, $b \in A_i$, and let $W_i^* = u_i(s)$. Since I assumed that $u_i(\cdot)$ is a vNM utility function, we have

$$W_i(b) = \sum_{c \in A_{-i}} u_i(b,c)s_{-i}(c) = \sum_{c \in A_{-i}} \bar{u}_i(b,c)s_{-i}(c) - \sum_{c \in A_{-i}} \frac{\sigma_i(b,c)s_{-i}(c)Z_i(b,c)}{\sqrt{n_i(b,c)}}.$$

Since $Z_i(\cdot)$ are i.i.d. with distribution $N(0,1)$,

$$\left(\sum_{c \in A_{-i}} \frac{\sigma_i(b,c)s_{-i}(c)Z_i(b,c)}{\sqrt{n_i(b,c)}}\right) \sim N\left(0, \sum_{c \in A_{-i}} \frac{\sigma_i^2(b,c)s_{-i}^2(c)}{n_i(b,c)}\right).$$

Then,

$$\left(\frac{\sum_{c \in A_{-i}} \left(\sigma_i(b,c)/\sqrt{n_i(b,c)}\right) s_{-i}(c)Z(b,c)}{\sqrt{\sum_{c \in A_{-i}} \frac{\sigma_i^2(b,c)s_{-i}^2(c)}{n_i(b,c)}}}\right) \sim N(0,1).$$

As a result, we can derive

$$\Pr(W_i(b) \leq u + \epsilon) = 1 - \Phi\left[\frac{\sum_{r \in Q_{-i}} \bar{u}_i(q,r)s_{-i}(r) - u - \epsilon}{\sqrt{\sum_{c \in A_{-i}} \frac{\sigma_i^2(b,c)s_{-i}^2(c)}{n_i(b,c)}}}\right]$$

Similarly we can show that

$$W_i^* \sim N(\sum_{a \in A} \bar{u}_i(a)s(a), \sum_{a \in A} \frac{\sigma^2(a)s^2(a)}{n_i(a)}).$$

We now get the desired result by applying Proposition 10.

A.6 Proof of Proposition 12

$$\Pr\left(r \text{ is } \epsilon\text{−dominant}\right) = \Pr\left(\max_{i \in I} \max_{t \in R_{-i}} \max_{r' \in R_i \setminus r_i} [u_i(r',t) - u_i(r_i,t)] \leq \epsilon\right) =$$
$$\prod_{i \in I} \prod_{t \in R_{-i}} \int_{\mathbb{R}} \prod_{r' \in R_i \setminus r_i} \Pr\{u_i(r',t) \leq u + \epsilon\} f_{u_i(r_i,t)}(u)du.$$

A.7 Proof of Proposition 13

$$\Pr\left(r \text{ is } \epsilon\text{−undominated}\right) = \Pr\left(\max_{i \in I} \min_{t \in R_{-i}} \max_{r' \in R_i \setminus r_i} [u_i(r',t) - u_i(r_i,t) \leq \epsilon]\right) =$$
$$\prod_{i \in I} \left[1 - \Pr\left(\max_{t \in R_{-i}} \min_{r' \in R_i \setminus r_i} [u_i(r',t) - u_i(r_i,t) > \epsilon]\right)\right].$$

Isolating the probability term, we get

$$\Pr\left(\max_{t\in R_{-i}}\min_{r'\in R_i\backslash r_i}[u_i(r',t)-u_i(r_i,t)>\epsilon]\right)=$$
$$\prod_{t\in R_{-i}}\int_{\mathbb{R}}\Pr\left(\min_{r'}[u_i(r',t)-u_i(r_i,t)>\epsilon]\right)f_{u_i}(u)du=$$
$$\prod_{t\in R_{-i}}\int_{\mathbb{R}}\left(1-\Pr\{\max_{r'}[u_i(r',t)-u_i(r_i,t)>\epsilon]\}\right\}f_{u_i}(u)du.$$

By substituting this expression into the one above, we obtain the desired result.

A.8 Proof of Proposition 15

$$\Pr([\max_{i\in I}\max_{t\in R_{-i}}(u_i(r)-u_i(r_i,t))\leq\delta]\&[\max_{i\in I}\max_{r'\in R_i}(u_i(r',r_{-i})-u_i(r))\leq\epsilon])=$$
$$\prod_{i\in I}\int_{\mathbb{R}}\Pr\{[\max_{t\in R_{-i}}u_i(r_i,t)\geq u-\delta]\&[\max_{r'\in R_i}u_i(r',r_{-i})\leq u+\epsilon]\}f_{u_i(r)}(u)du.$$

The result then follows from independence.

A.9 Proof of Proposition 16

$$\Pr\{\max_{i\in I}\max_{t\in R_{-i}}[u_i(r)-u_i(r_i,t)]\leq\delta\ \&\max_{i\in I}\max_{t\in R_{-i}}\max_{r'\in R_i}[u_i(r',t)-u_i(r_i,t)]\leq\epsilon\}=$$
$$\prod_{i\in I}\int_{\mathbb{R}}\prod_{t\in R_{-i}}\Pr\{[u_i(r_i,t)\geq u-\delta]\ \&[\max_{r'\in R_i}[u_i(r',t)-u_i(r_i,t)]\leq\epsilon]\}f_{u_i(r)}du=$$
$$\prod_{i\in I}\int_{\mathbb{R}}\prod_{t\in R_{-i}}\Pr\{u_i(r_i,t)\geq u-\delta\}\times\Pr\{\max_{r'\in R_i}[u_i(r',t)-u_i(r_i,t)]\leq\epsilon|u_i(r_i,t)\geq u-\delta\}f_{u_i(r)}du.$$

Since

$$\Pr\{\max_{r'\in R_i}[u_i(r',t)-u_i(r_i,t)]\leq\epsilon|u_i(r_i,t)\geq u-\delta\}\geq\prod_{r'\in R_i}\Pr\{u_i(r',t)\leq u+\epsilon-\delta\},$$

the result follows.

A.10 Proof of Proposition 17

$$\Pr\{\max_{i\in I}\max_{t\in R_{-i}\backslash r_{-i}}[u_i(r)-u_i(r_i,t)]\le\delta \;\&\max_{i\in I}\min_{t\in R_{-i}}\max_{r'\in R_i\backslash r_i}[u_i(r',t)-u_i(r_i,t)]\le\epsilon\}=$$
$$\prod_{i\in I}\int_{\mathbb{R}}\Bigg[\Pr\{\max_{t\in R_{-i}\backslash r_{-i}}u_i(r_i,t)\ge u-\delta\;\bigcap$$
$$\left(\min_{t\in R_{-i}\backslash r_{-i}}\max_{r'\in R_i\backslash r_i}[u_i(r',t)-u_i(r_i,t)]\le\epsilon\bigcup\max_{r'\in R_i\backslash r_i}u_i(r',r_{-i})\le\epsilon+u\right)\}\Bigg]f_{u_i(r)}(u)du=$$
$$\prod_{i\in I}\int_{\mathbb{R}}\Bigg[\Pr\{\max_{t\in R_{-i}\backslash r_{-i}}[u_i(r_i,t)]\ge u-\delta\}\times$$
$$\times\Pr\{\min_{t\in R_{-i}\backslash r_{-i}}\max_{r'\in R_i\backslash r_i}[u_i(r',t)-u_i(r_i,t)]\le\epsilon\bigcup$$
$$\max_{r'\in R_i\backslash r_i}u_i(r',r_{-i})\le\epsilon+u\mid\max_{t\in R_{-i}\backslash r_{-i}}u_i(r_i,t)\ge u-\delta\}\Bigg]f_{u_i(r)}(u)du.$$

Now, let's isolate each of the multiplicand $\Pr\{\cdot\}$ parts above. The first is equivalent to

$$\prod_{t\in R_{-i}\backslash r_{-i}}\Pr\{u_i(r_i,t)\ge u-\delta\}.$$

The case of the second is more complicated:

$$\Pr\{\min_{t\in R_{-i}\backslash r_{-i}}\max_{r'\in R_i\backslash r_i}[u_i(r',t)-u_i(r_i,t)]\le\epsilon\bigcup$$
$$\max_{r'\in R_i\backslash r_i}u_i(r',r_{-i})\le\epsilon+u\mid\max_{t\in R_{-i}\backslash r_{-i}}u_i(r_i,t)\ge u-\delta\}=$$
$$1-\Pr\{\max_{t\in R_{-i}\backslash r_{-i}}\min_{r'\in R_i\backslash r_i}[u_i(r',t)-u_i(r_i,t)]>\epsilon\mid\max_{t\in R_{-i}\backslash r_{-i}}u_i(r_i,t)\ge u-\delta\}\times$$
$$\times\Pr\{\min_{r'\in R_i\backslash r_i}u_i(r',r_{-i})>\epsilon+u\}.$$

Again, isolating the two multiplicands for convenience, we have

$$1-\prod_{r'\in R_i\backslash r_i}\Pr\{u_i(r',r_{-i})>\epsilon+u\}$$

for the second. As for the first,

$$\Pr\{\max_{t\in R_{-i}\backslash r_{-i}}\min_{r'\in R_i\backslash r_i}[u_i(r',t)-u_i(r_i,t)]>\epsilon\mid\max_{t\in R_{-i}\backslash r_{-i}}u_i(r_i,t)\ge u-\delta\}=$$
$$\prod_{t\in R_{-i}\backslash r_{-i}}\Pr\{\min_{r'\in R_i\backslash r_i}[u_i(r',t)-u_i(r_i,t)]>\epsilon\mid u_i(r_i,t)\ge u-\delta\}.$$

At this point, I will proceed by first proving the following claim:

Claim. *Let X,Y be random variables, A,B Borel sets, and suppose we have probability measures for X,Y, as well as the corresponding product measure (I designate each simply using $\Pr\{\cdot\}$, endowing it with the appropriate probability measure). Then*

$$\Pr\{f(X,Y)\in A|Y\in B\}=\frac{\int_B\Pr\{f(X,y)\in A\}f_Y(y)dy}{\int_B f_Y(y)dy},$$

where $F_Y(y)$ is the distribution function for Y.

To prove this, first observe that by Bayes rule,

$$\Pr\{f(X,Y) \in A | Y \in B\} = \frac{\Pr\{f(X,Y) \in A \bigcap Y \in B\}}{\Pr\{Y \in B\}}.$$

For the numerator, I apply smoothing with respect to Y to get

$$\begin{aligned} &\Pr\{f(X,Y) \in A \bigcap Y \in B\} = \\ &\quad E_Y[\Pr\{f(X,Y) \in A \bigcap Y \in B\} | Y = y] = \\ &\quad \Pr\{f(X,y) \in A \bigcap y \in B\} = \\ &\quad \int_{\mathbb{R}} \Pr\{f(X,y) \in A \bigcap y \in B\} f_Y(y) dy. \end{aligned}$$

Since

$$\Pr\{f(X,y) \in A \bigcap y \in B\} = \begin{cases} \Pr\{f(X,y) \in A\} & \text{if } y \in B, \\ 0 & \text{if } y \notin B \end{cases}$$

the claim follows.

Now I apply the claim to get

$$\begin{aligned} &\Pr\{\min_{r' \in R_i \backslash r_i} [u_i(r',t) - u_i(r_i,t)] > \epsilon \, | u_i(r_i,t) \geq u - \delta\} = \\ &\frac{\int_{u-\delta}^{\infty} \Pr\{\min_{r' \in R_i \backslash r_i} u_i(r',t) > \epsilon + v\} f_{u_i(r_i,t)}(v) dv}{\int_{u-\delta}^{\infty} f_{u_i(r_i,t)}(v) dv}. \end{aligned}$$

Finally,

$$\Pr\{\min_{r' \in R_i \backslash r_i} u_i(r',t) > \epsilon + v\} = \prod_{r' \in R_i \backslash r_i} [1 - \Pr\{u_i(r',t) \leq \epsilon + v\}].$$

Combining all the pieces gives us the desired result.

A.11 Proof of Proposition 18

$$\begin{aligned} \Pr\left(\max_{i \in I} \sup_{t \in R_i} u_i(t, r_{-i}) - u_i(r) \leq \epsilon\right) \geq \Pr\left(\max_{i \in I} \max_{t \in R_{k,i}} u_i(t, r_{-i}) - u_i(r) \leq \epsilon - Bd\right) = \\ = \prod_{i \in I} \int_{\mathbb{R}} \prod_{t \in R_{k,i}} \Pr(u_i(t, r_{-i}) \leq u + \epsilon - Bd) f_{u_i(r)}(u) du. \end{aligned}$$