Analyzing Complex Strategic Interactions in Multi-Agent Systems

William E. Walsh Rajarshi Das Gerald Tesauro Jeffrey O. Kephart
IBM T.J. Watson Research Center
19 Skyline Drive, Hawthorne, NY 10532, USA
{wwalsh1, rajarshi, gtesauro, kephart}@us.ibm.com

Abstract

We develop a model for analyzing complex games with repeated interactions, for which a full game-theoretic analysis is intractable. Our approach treats exogenously specified, heuristic strategies, rather than the atomic actions, as primitive, and computes a heuristic-payoff table specifying the expected payoffs of the joint heuristic strategy space. We analyze two games based on (i) automated dynamic pricing and (ii) continuous double auction. For each game we compute Nash equilibria of previously published heuristic strategies. To determine the most plausible equilibria, we study the replicator dynamics of a large population playing the strategies. In order to account for errors in estimation of payoffs or improvements in strategies, we also analyze the dynamics and equilibria based on perturbed payoffs.

Introduction

Understanding complex strategic interactions in multi-agent systems is assuming an ever-greater importance. In the realm of agent-mediated electronic commerce, for example, authors have recently discussed scenarios in which self-interested software agents execute various dynamic pricing strategies, including posted pricing, bilateral negotiation, and bidding. Understanding the interactions among various strategies can be extremely valuable, both to designers of markets (who wish to ensure economic efficiency and stability) and to designers of individual agents (who wish to find strategies that maximize profits). More generally, by demystifying strategic interactions among agents, we can improve our ability to predict (and therefore design) the overall behavior of multi-agent systems—thus reducing one of the canonical pitfalls of agent-oriented programming (Jennings & Wooldridge 2002).

In principle, the (Bayes) Nash equilibrium is an appropriate concept for understanding and characterizing the strategic behavior of systems of self-interested agents. In practice, however, it is infeasible to compute Nash equilibria for all but the very simplest interactions. For some types of repeated interactions, such as continuous double auctions (Rust, Miller, & Palmer 1993) and simultaneous ascending auctions (Milgrom 2000), even formulating the information structure of the extensive-form game, much less computing the equilibrium, remains an unsolved problem.

Given this state of affairs, it is typical to endow agents with heuristic strategies, comprising hand-crafted or learned decision rules on the underlying primitive actions as a function of the information available to an agent. Some strategies are justified on the basis of desirable properties that can be proven in simplified or special-case models, while others are based on a combination of economic intuition and engineering experience (Greenwald & Stone 2001; Tesauro & Das 2001).

In this paper, we propose a methodology for analyzing complex strategic interactions based on high-level, heuristic strategies. The core analytical components of our methodology are Nash equilibrium of the heuristic strategies, dynamics of equilibrium convergence, and perturbation analysis.

Equilibrium and the dynamics of equilibrium convergence have been widely studied, and our adoption of these tools is by no means unique. Yet, these approaches have not been widely and fully applied to the analysis of heuristic strategies. A typical approach to evaluating the strategies has been to compare various mixtures of strategies in a structured or evolutionary tournament (Axelrod 1997; Rust, Miller, & Palmer 1993; Wellman *et al.* 2001), often with the goal of establishing which strategy is the "best". Sometimes, the answer is well-defined, as in the Year 2000 Trading Agent Competition (TAC-00), in which the top several strategies were quite similar, and clearly superior to all other strategies (Greenwald & Stone 2001). In other cases, including recent studies of strategies in continuous double auctions (Tesauro & Das 2001) and in TAC-01, there does not appear to be any one dominant strategy.

The question of which strategy is "best" is often not the most appropriate, given that a *mix* of strategies may constitute an equilibrium. The tournament approach itself is often unsatisfactory because it cannot easily provide a complete understanding of multi-agent strategic interactions, since the tournament play is just one trajectory through an essentially infinite space of possible interactions. One can never be certain that all possible modes of collective behavior have been explored.

Our approach is a more principled and complete method for analyzing the interactions among heterogeneous heuristic strategies. Our methodology, described more fully in the following sections, entails creating a heuristic-payoff table—an analog of the usual payoff table, except that the entries describe expected payoffs for high-level, heuristic strategies rather than primitive actions. The heuristic-payoff table is then used as the basis for several forms of analysis.

The next four sections of our paper, "Modeling Approach", "Equilibrium Computation", "Dynamic Analysis", and "Perturbation of Payoffs" detail the components of our approach. After presenting the methodology, we describe its application to two complex multi-agent games: automated dynamic pricing (ADP) in a competing-seller scenario, and automated bidding in the continuous double auction (CDA). We conclude with a discussion of the methodology.

Modeling Approach

We start with a game that may include complex, repeated interactions between A agents. The underlying rules of the game are well-specified and common knowledge. The rules specify particular ***actions*** that agents may take as a function of the state of the game. Each of the A agents has a choice of the same S exogenously specified, heuristic strategies. By ***strategy***, we mean a policy that governs the choice of individual actions, typically expressed as a deterministic or stochastic mapping from the information available to the agent to an action. For example in the CDA, typical actions are of the form "bid b at time t", while the bidding strategies can be complex functions, expressed in hundreds or thousands of lines of code, that specify what bids are placed over the course of trading. We say the strategies are "heuristic" in that they are not generally the solution of (Bayes) Nash equilibrium analysis.

A key step in our methodology is to compute a ***heuristic-payoff table***, that specifies the expected payoff to each agent as a function of the strategies played by all the agents. The underlying payoffs may depend on varying types for the agents, which may encompass, for instance, different utility functions or different roles. To simplify analysis, we assume that the types of agents are drawn independently from the same distribution, as is common in the auction literature. To further improve tractability, we also assume that that an agent chooses its strategy independently of its type. However, we do assume that the agent's type and the distribution of types are available to the strategy itself.

The heuristic-payoff table is an abstract representation of the fundamental game in that we have reduced the model of the game from a potentially very complex, multi-stage game to a one-shot game in normal form, in which we treat the choice of heuristic strategies, rather than the basic actions, as the level of decision making for strategic analysis. We emphasize that we take the strategies as exogenous and do not directly analyze their genesis nor their composition. With this model, we can apply the standard game-theoretic analysis just as we would with a normal-form game of simple actions, such as the prisoner's dilemma.

The standard payoff table for a normal-form game requires S^A entries, which can be extremely large, even when S and A are moderate. But we have restricted our analysis to symmetric games in which each agent has the same set of strategies and the same distribution of types (and hence payoffs). Hence, we can merely compute the payoff for each strategy as a function of the *number* of agents playing each strategy, without being concerned about the individual identities of those agents. This symmetry reduces the size of the payoff table enormously. The number of entries in the table is the number of unique ways into which a set of A identical agents may be partitioned among S strategies. This quantity can be shown to be $\binom{A+S-1}{A}$, or roughly $\frac{A^{S-1}}{(S-1)!}$ when $A \gg S$. In this case, changing the exponent from A to $S-1$ results in a huge reduction in the size of the payoff table. For example, in applications presented in this paper, $A = 20$, $S = 3$, and thus the symmetric payoff table contains just 231 entries—far less than the approximately 3.48×10^9 entries contained in the asymmetric payoff table.

Payoffs may be computed analytically for sufficiently simple games and strategies. However, in games of realistic complexity, it is necessary to measure expected payoffs in simulation. In games with non-deterministic aspects, it will be necessary to run many independent simulations for each payoff table entry, to ensure that the measured payoffs are as accurate as possible.

With a payoff table in hand, one can use a variety of techniques to gain insight into a system's collective behavior. We present three such techniques in this paper: (1) First, we perform a static analysis, which entails computing Nash equilibria of the payoff table. To simplify this computation, we look only for symmetric Nash equilibria, although generally nothing precludes the existence of asymmetric equilibria. (2) Second, we model the dynamics of agents that periodically and asynchronously switch strategies to ones that appear to be more successful. This helps us understand the evolution of the mixture of strategies within the population, providing insight into the conditions under which of the various Nash equilibria may be realized. (3) Third, we suggest techniques for understanding the strategies themselves at a deeper level. Since our analysis is performed at the more coarse-grained level of strategies rather than actions, it does not directly provide insight into the behavior of the strategies at the level of primitive actions. We would like to understand how to design good strategies, how the series of primitive actions interact between strategies, how to improve strategies, and the system-level effects of strategy improvements. We have begun to address these challenges in two ways. We analyze how the static and dynamic properties of the system change when the values in the payoff table are perturbed. Perturbation analysis suggests the potential effects of improving one of the strategies, as well as noise or measurement errors in the payoffs. All of these techniques will be elucidated in the following three sections.

Comments on Our Modeling Approach

Obtaining the payoff table is the key step in our methodology. Given the payoff table, we need not concern ourselves any more with the details of how it was generated; the rest of the analyses that we perform all operate mechanically on the table values, with no need to re-do the simulations that generated them. Still, obtaining the payoff table at a high degree of precision may be computationally non-trivial. A 20 agent, 3 strategy game requires the computation of 231 table entries. In our analysis of the CDA game, we averaged the results of 2500 simulations for each payoff entry, requiring about an afternoon of computation on seven 333MHz-450MHz IBM RS6000 workstations to compute the entire table. Computing the payoff table for our application was feasible because the auctions and heuristic strategies require fairly simple computations and we had sufficient computational resources available.

For some games (e.g., iterative combinatorial auctions) and strategies (e.g., that use expensive optimization algorithms), computing the payoff table may be exorbitantly expensive. Other factors, such as limited resources or timing properties of the game, may limit our ability to fully compute the payoff table. For instance, each TAC-01 game takes exactly 15 minutes, and only two auction servers are publicly available at present. Clearly, for our methodology to be more broadly feasible, the expense of computing the payoff table must be addressed. Currently, we are exploring ways to interleave Nash equilibrium computation with payoff computation. We believe that this will admit methods to help focus on the payoff entries where accuracy is most needed, rather than performing a large number of simulations for all entries.

Our methodology is based on heuristic strategies to address the limitations in present game-theoretic techniques. On the other hand, our assumptions of symmetry are made strictly to reduce storage and computational cost. While it is straightforward *in principle* to break the symmetry of type distributions and strategy sets, the exponential growth in table size and computation, both for determining the payoff entries and for the equilibrium and dynamic analysis we discuss in subsequent sections, practically restricts the degree to which we can break symmetry.

We acknowledge that, despite its expedience, the most controversial aspect of our approach is likely to be the assumption of a single set of heuristic strategies available to all agents, which avoids the issue of how the strategies are developed and become known to the agents. Nevertheless, an observation of current practice suggests that a moderately sized set of heuristic strategies can become available to the population at large. Two automated bidding services are openly available to users of eBay, and investing in a mutual fund is an exact method of adopting the fund manager's strategy. Indeed, as automated negotiation becomes more widespread and pervasive, we expect that the majority of participants in the general public will adopt heuristic strategies developed by others, rather than developing their own. Admittedly, some participants with sufficient talent and resources will invest in the development of highly specialized strategies. But where there is an information flow, there is the possibility for these strategies to spread. Many players in TAC-01 used techniques publicly documented and shown to be effective in TAC-00.

Another restriction we have made in our model is the assumption that agents make their choices of strategy independently of their own type. We should generally expect that sophisticated agents would choose to condition their strategy choice based on their type.[1] While we could easily model type conditioning by discretizing types, the storage and computational expense quickly explodes, as with computing asymmetric equilibria.

Equilibrium Computation

At the start of the game, each of the A agents chooses to play one of the S available pure strategies. The payoff to agent i is a real-valued function u of the strategy played by i and the strategies played by all other agents. As discussed in the modeling section, the payoff is the expected reward obtained when the agents play a particular combination of strategies. Because we assume symmetric strategy sets and payoffs, the payoff to an agent can be represented as the payoff to each strategy as a function of the number of agents playing each strategy.

Agent i may choose its strategies randomly according to a ***mixed strategy***, $\hat{p}_i = (\hat{p}_{i,1}, \ldots, \hat{p}_{i,S})$. Here, $\hat{p}_{i,j}$ indicates the probability that agent i plays strategy j, with the constraints that $\hat{p}_{i,j} \in [0,1]$ and $\sum_{j=1}^{S} \hat{p}_{i,j} = 1$. The vector of all agents' mixed strategies is $\hat{p}$ and the vector of mixed strategies for all agents except i is $\hat{p}_{-i}$. We indicate by $\hat{p}_i = e^j$, the special case when agent i plays pure strategy j with probability one.

We denote by $u(e^j, \hat{p}_{-i})$ the expected payoff to an agent i for playing pure strategy j, given that all other agents play their mixed strategies $\hat{p}_{-i}$. The expected payoff to agent i of the mixed strategy is then $u(\hat{p}_i, \hat{p}_{i-1}) = \sum_{j=1}^{S} u(e^j, \hat{p}_{i-1}) \hat{p}_{i,j}$.

In game theoretic analysis, it is generally assumed that rational agents would play mixed Nash equilibrium strategies, whereby no one agent can receive a higher payoff by unilaterally deviating to another strategy, given fixed opponents' strategies. Formally, probabilities $\hat{p}^*$ constitute a ***Nash equilibrium*** iff for all agents i, and all $\hat{p}_i$, $u(\hat{p}_i, \hat{p}^*_{-i}) \leq u(\hat{p}^*_i, \hat{p}^*_{-i})$.

In the remainder of this paper, we restrict our attention to symmetric mixed strategy equilibria, whereby $\hat{p}^*_i = \hat{p}^*_k = p^*$ for all agents i and k. We denote an arbitrary, symmetric, mixed strategy by p and the probability that a given agent plays pure strategy j by p_j. Nash equilibria of symmetric strategies always exist for symmetric games (Weibull 1995), and are not generally unique. We restrict our attention in this way for two reasons. First, when searching for symmetric Nash equilibria, we need only find S, rather than AS probabilities. Second, absent a particular mechanism for breaking symmetry, it is reasonable to assume symmetry by default, as is often done in auction analysis with symmetric agent types. In particular, symmetry is consistent with the particular evolutionary game theory model we consider in the next section.

Finding Nash equilibria can be a computationally challenging problem, requiring solutions to complex, nonlinear equations in the general case. The Nash equilibrium conditions can be expressed in various equivalent formulations, each suggesting different solution methods (McKelvey & McLennan 1996). Several solution methods are

[1] Although an individual agent will always do as well or better to condition on type, given the strategic behavior of the other agents, an equilibrium of strategies conditioned on type could actually be worse for all agents than a non-type-conditioned equilibrium.

implemented in Gambit (McKelvey, McLennan, & Turocy 2000), a freely available game solver. But because Gambit is not able to exploit the symmetry of the games we study, it requires the full normal form game table as input, severely limiting the size of problems that can feasibly be represented in the program.

In this work, we formulate Nash equilibrium as a minimum of a function on a polytope. Restricting ourselves to symmetric equilibria in a symmetric game, the problem is to minimize:

$$v(p) = \sum_{j=1}^{S} (\max[u(e^j, p) - u(p, p),\, 0])^2 \qquad (1)$$

The mixed strategy p^* is a Nash equilibrium iff it is a global minimum of v (McKelvey & McLennan 1996). Although not all minima of the function v may be global, we can validate that a minimum is global if its value is zero.

In this work we used amoeba (Press *et al.* 1992), a nonlinear optimizer, to find the zero-points of v in our application games. Amoeba searches an n-dimensional space using a $(n+1)$-dimensional simplex or polyhedron. The function is evaluated at each vertex of the simplex and the polyhedron attempts to move down the estimated gradient by a series of geometric transformations that continually strive to replace the worst-performing vertex. We repeatedly ran amoeba, restarting at random points on the S-dimensional simplex, and stopping when it had found 30 previously-discovered equilibria in a row. Because $S = 3$ in our applications, we were able to plot v and verify that we had indeed found all equilibria. For games with $A = 20$, $S = 3$, and three equilibria, it took amoeba roughly ten minutes to terminate on a 450Mhz IBM RS6000 machine.

Dynamic Analysis

Nash equilibria provide a theoretically satisfying view of ideal static properties of a multi-agent system. Yet often the dynamic properties may be of equal or greater concern. In actual systems, it may be unreasonable to assume that agents have all correct, common knowledge necessary to compute equilibria. Furthermore, even when agents have this common knowledge and the resources to compute equilibria, we still want to address the question of which equilibrium is chosen and how agents (implicitly) coordinate to reach it.

Many have studied models of adaptation and learning to simultaneously address these issues. Common learning approaches adjust strategy mixes gradually to myopically improve payoffs in repeated play of a game. Definitive, positive theoretical properties of equilibrium convergence for general games of A agents and S strategies, have not yet been established (Fudenberg & Kreps 1993; Fudenberg & Levine 1993; Jordan 1993). For two-player, two-strategy games, iterated Gradient Ascent (Singh, Kearns, & Mansour 2000) provably generates dynamics giving an average payoff equivalent to some Nash equilibrium. Greenwald and Kephart (2001) observe empirically that agents using methods based on no external regret (Freund & Schapire 1996) and no internal regret (Foster & Vohra 1997) learning play pure strategies in a frequency corresponding to a Nash equilibrium. Approaches based on Q-Learning (Watkins 1989) optimize long-term payoffs rather than only next-stage payoffs. Equilibrium convergence can be guaranteed for a single Q-Learner, and for two-player games in the zero-sum case (Littman 1994), or in the general-sum case with the use of a Nash equilibrium solver (Hu & Wellman 1998).

For this paper, we borrow a well-developed model from evolutionary game theory (Weibull 1995) to analyze strategy choice dynamics. In contrast to the aforementioned approaches, which model repeated interactions of the same set of players (i.e., the game players constitute the population), we posit a large population of N agents, from which $A \ll N$ agents are randomly selected at each time step to play the game. At any given time, each agent in the population plays one of the S pure strategies, and the fraction of agents playing strategy j is p_j.[2] The p_j values define a population vector of strategy shares p. For sufficiently large N, p_j may be treated as a continuous variable.

We use the ***replicator dynamics*** formalism to model the evolution of p with time as follows:

$$\dot{p}_j = [u(e^j, p) - u(p, p)]p_j \qquad (2)$$

where $u(p, p)$ is the population average payoff, and $u(e^j, p)$ is the average payoff to agents currently using pure strategy j. Equation 2 models the tendency of strategies with greater than average payoff to attract more followers, and strategies with less than average payoff to suffer defections.

We prefer that a dynamic model assume minimal informational requirements for agents beyond their own actions and payoffs. The replicator dynamics equation implies that agents know $u(p, p)$, a rather implausible level of information. However, we can obtain the same population dynamics with a more plausible "replication by imitation" model (Weibull 1995). In that model, an agent switches to the strategy of a randomly chosen opponent who appears to be receiving a higher payoff. Alternative models in which learning at the individual level leads to replicator dynamics has been discussed in (Borgers & Sarin 1997).

We could interpret p, at a given time, as representing a symmetric mixed strategy for all N players in the game. With this interpretation, the fixed points of Equation 2 (where $\dot{p}_j = 0$ for all strategies j), correspond to Nash equilibria, and $u(p, p)$ and $u(e^j, p)$ are as defined in the equilibrium context. When strategy trajectories governed by Equation 2 converge to an equilibrium, the equilibrium is an ***attractor***. However, these strategy trajectories do not necessarily terminate at fixed points. Indeed, there are many plausible payoff functions that generate limit cycle trajectories (Weibull 1995) or even chaotic trajectories (Sato, Akiyama, & Farmer 2001).

When multiple Nash equilibria exist, those that are attractors are clearly the only plausible equilibria within the evolutionary model. With multiple attractors, those with larger basins of attraction are more likely, assuming that every initial population state is equally likely. Alternatively, we can use the basins of attraction to understand which initial population mixes will lead to which equilibrium. Strategy designers who have an interest (e.g., fame or profits for selling

[2]Our motivation for overloading the p notation will become evident below.

software implementing the strategy) in widespread adoption of their strategies could then determine how much initial adoption is necessary to lead to an equilibrium containing a favorable ratio of their strategies.

For our analysis of two particular games (in the "Applications" section) we use the heuristic payoff table and Equation 2 to generate a large number of strategy trajectories, starting from a broad distribution of initial strategy vectors p. For three strategy choices, the resulting flows can be plotted in a two-dimensional unit simplex and have an immediate visual interpretation.

Perturbation of Payoffs

In our model, we have assumed a fixed set of exogenously specified strategies. But because they are heuristic, rather than game-theoretically computed, we should generally assume that there could be many variations that engender changes in performance. Because the possible variations could potentially be infinite, we do not have a way to account for all variations with our methodology. Still, by perturbing the payoff table in some meaningful ways, we can perform some directed study of plausible effects of certain abstract changes in strategy behavior.

A question we may ask is how would improving one strategy relative to the others affect the equilibria and dynamics. We consider a simple model in which the agents playing strategy σ^+ "steal" some fraction $\alpha \in [0,1]$ of the payoff from the agents playing the other strategies. For each profile of strategies in the payoff table, and each strategy $\sigma^- \neq \sigma^+$, where n^+ agents play σ^+, n^- agents play σ^-, and $q(\sigma^+)$ is the payoff of strategy σ^+, we change the payoffs as follows:

$$q(\sigma^+) = (1 + \alpha \min(n^+, n^-)/n^+) q(\sigma^+)$$
$$q(\sigma^-) = (1 - \alpha \min(n^+, n^-)/n^-) q(\sigma^-).$$

Note that, for any profile, this perturbation conserves the total payoff to all agents.

While it may not actually be possible to uniformly improve a strategy as we describe, the approach is suggestive of the type of perturbation that we might reasonably consider. Other possibilities could include taking surplus from only one opponent strategy, or uniform improvements to all strategies (which might occur if one strategy becomes more adept at executing win-win actions). These kinds of perturbations could help direct efforts to improve a strategy's likelihood of being adopted and the payoffs in equilibrium.

Alternatively, we might be interested in estimating the effects of unmodelled variations in the strategies throughout the population, or performing sensitivity analysis on the estimates of expected payoffs. To do so, we could perturb individual payoff entries randomly, either statically or whenever used in Equation (2).

Applications

In this section we apply our methodology to two games with complex, strategic interactions: an Automated Dynamic Pricing (ADP) game, and a Continuous Double Auction (CDA) game. We chose these games because of the intractability of computing equilibria in the underlying games, an existing body of literature which includes interesting heuristic strategies, and the availability of simulators for computing the heuristic payoff tables.

Automated Dynamic Pricing Game

Description of the ADP Game Interest in Internet commerce has fueled the emergence of software agents such as *shopbots* that greatly facilitate comparison shopping by buyers. Shopbots may also enable seller agents called *pricebots* to dynamically set posted prices based on competitor behavior. An example is the site `buy.com`, which monitors its primary competitors' prices and automatically undercuts the lowest. Interactions among such pricebots can generate rather complex price dynamics.

Models of interacting pricebots using a variety of pricing strategies have been studied in (Greenwald & Kephart 1999; Greenwald, Kephart, & Tesauro 1999). In these models, a set of sellers offers a single homogeneous good to a much larger set of buyers. At random times, over the course of a large number of discrete time steps, sellers reset their prices and buyers attempt to purchase. A buyer wishes to purchase one unit of the good, preferring to pay lower prices not exceeding its value, which is randomly chosen from the uniform distribution in the unit interval. All sellers have the same constant production cost, and their objective is to maximize the product of their per-unit profit and the number of sales. We assume that buyers use one of two simple rules for seller selection. A fixed 50% of buyers selects a seller at random, and the rest use a shopbot to find the current lowest-price seller.

We formulate a one-shot ADP Game of heuristic strategies, abstracting the underlying game of repeated price-setting by sellers. At the start of the game, sellers choose one of three heuristic dynamic pricing strategies to use for the duration of the game. The "game theory" (GT) strategy (Greenwald & Kephart 1999) plays a mixed-strategy Nash equilibrium computed for the underlying game assuming that all pricing and purchasing decisions are made simultaneously. The "derivative-follower" (DF) strategy (Greenwald & Kephart 1999) implements a simple hill-climbing adaptation, experimenting with incremental price adjustments to improve observed profitability, while ignoring assumptions about buyers or competitors. The "No-Internal-Regret" (NIR) strategy (Greenwald & Kephart 2001) adapts learning techniques from Foster Vohra (1997) to adaptively improve its pricing.

Following the simulation procedures in (Greenwald & Kephart 1999; 2001), we computed heuristic payoff tables for seller population sizes of 5 and 20 pricebots. Each table entry indicates the time-averaged payoff over 1 million time steps, with a single seller resetting its price at each time step.

Analysis of ADP Game Table 1 shows the Nash equilibria for ADP Games with 5 and 20 pricebots. None of the equilibria involve the DF pricing strategy, signifying its relative weakness. Among the equilibria in Table 1 only **A** is a pure-strategy Nash equilibrium. It is also interesting to note that the number of equilibria dwindles from three to one as the size of the ADP Game increases from 5 to 20 pricebots.

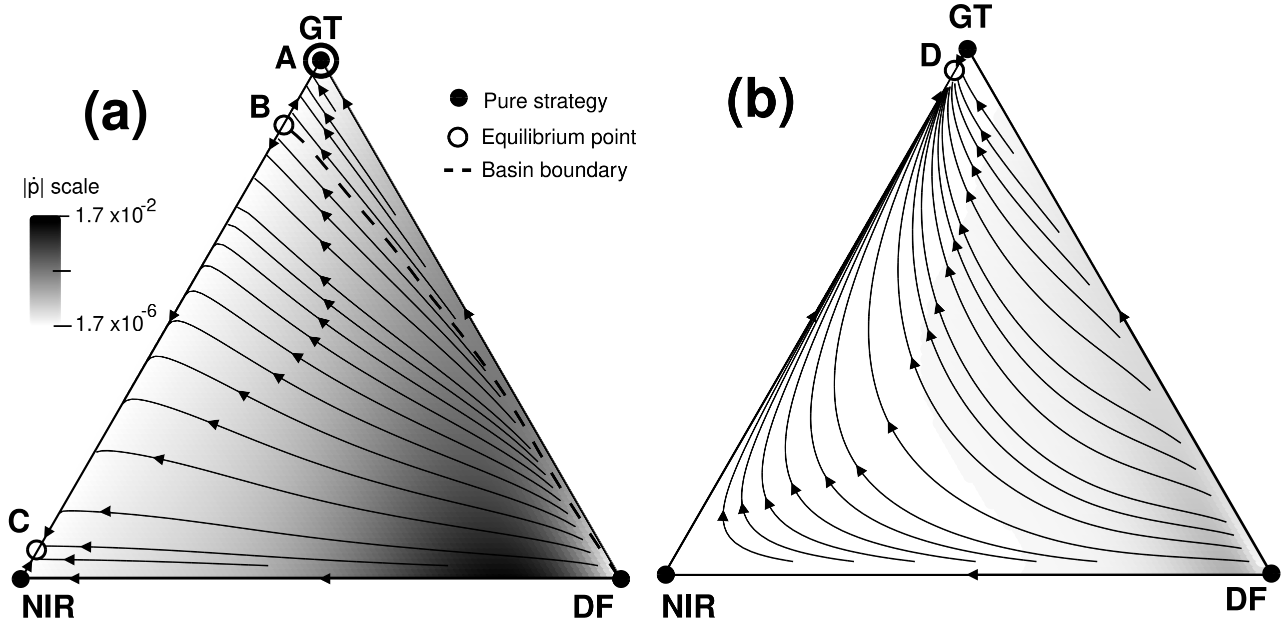

Figure 1: (a) Replicator dynamics for the Automated Pricing Game with 5 pricebots. Points p in the simplex represent strategy mixes, with homogeneous populations labeled at the vertices. The trajectories in the simplex describe the motion of p following Equation 2. Open circles are Nash equilibria, corresponding to fixed points of Equation 2, with labels corresponding to those in Table 1. The dashed line denotes the boundary of the two basins of attraction. The gray shading is proportional to the magnitude of $\dot{p}$. (b) Replicator dynamics for the Automated Pricing Game with 20 pricebots.

Agents	Label	p(GT)	p(DF)	p(NIR)	Payoff
5	**A**	1.000	0.000	0.000	0.051
	B	0.871	0.000	0.129	0.049
	C	0.030	0.000	0.969	0.047
20	**D**	0.986	0.000	0.014	0.013

Table 1: The symmetric Nash mixed-strategy equilibria for the Automated Pricing Game with 5 and 20 pricebots. Each row is an equilibrium, showing the probabilities of choosing the high-level strategies (GT, DF, and NIR), and the expected equilibrium payoff. The labels of the equilibria correspond to those shown in Figures 1(a) and 1(b).

The replicator dynamics for the ADP Game are shown in Figure 1(a) and 1(b). The strategy space is represented by a two-dimensional unit simplex with vertices corresponding to the pure strategies $p = (1,0,0)$ (all GT), $p = (0,1,0)$ (all DF), and $p = (0,0,1)$ (all NIR). Trajectories are obtained by starting from an initial point and applying Equation 2 repeatedly until $\dot{p} = 0$.

For the 5-agent case shown in Figure 1(a), points **A**, **B**, and **C** are the Nash equilibria shown in Table 1. **A** and **C** are attractors, while **B** is a saddle point. Therefore only **A** and **C** can be reached asymptotically as **B** is unstable to small fluctuations. Note that **C**, consisting almost entirely of the NIR strategy, has a much larger basin of attraction than the pure-GT point **A**, suggesting it to be the most likely outcome. Although it is not correct to refer to NIR as the "best" strategy, its attraction is the strongest in the 5-agent game.

In the 20-agent case shown in Figure 1(b), however, we find a surprising reversal: there is now only one Nash equilibrium, consisting almost entirely of GT. NIR is much weaker, relative to GT, as compared to the 5-agent game. NIR nevertheless can play a crucial role in determining the global flow in the strategy simplex. While all trajectories terminate at **D**, a significant fraction of them pass close to the pure-NIR vertex. The light shading in this vicinity indicates a very small magnitude of $\dot{p}$. This implies that even though all-NIR is not a Nash equilibrium, the population can spend a significantly long time with most agents adopting the NIR strategy.

Perturbation analysis of the population dynamics, with 20 sellers, $\sigma^+ = $ NIR, $\sigma^- = $ GT and $\alpha \approx 0.06$, results in the emergence of two new Nash equilibrium points (one an attractor and another unstable), consisting solely of a mix of GT and NIR strategies. When $\alpha \approx 0.0675$, there is an attractor equilibrium with a majority of NIR and a basin of attraction covering more than half of the simplex. Further increasing α progressively decreases **D**'s basin of attraction. By the point at which α increases to ≈ 0.1, **D** disappears completely and a single Nash equilibrium point remains near the pure strategy NIR vertex. The resulting simplex flow is similar to that in Figure 1(b), but with the positions of the pure strategy vertices GT and NIR interchanged. In short, NIR would start becoming a strong strategy with a 6.75% improvement, and would become nearly dominant with a 10% improvement.

The weakness of NIR in the 20-agent game was quite unexpected to us, given the strength it exhibited in a previous study using up to five agents (Greenwald & Kephart 2001), and in the present 5-agent game. This demonstrates the value of performing equilibrium and dynamic analysis on a relatively large number of agents. The results here have suggested to us a number of avenues for deeper study and development of the strategies, further demonstrating the value of our methodology. Perhaps NIR, which has several parameters, needs to be retuned for populations of different sizes. Alternatively, our results may show that it is more difficult for NIR to learn when playing against a greater number of agents. Whatever conclusions could be reached, we have already gained a greater understanding with our methodology than we could with a more simple analysis.

Continuous Double Auction Game

Description of the CDA Game The CDA is the predominant mechanism used for trading in financial markets such as NASDAQ and NYSE. A CDA continually accepts bids, which either immediately match pending bids, or remain standing until matched by later bids. Models of CDAs have been extensively studied using both human traders (Smith 1982) and computerized traders (Rust, Miller, & Palmer 1993; Cliff & Bruten 1997; Gjerstad & Dickhaut 1998). Based on these, we adopt a model in which agents trade in a CDA marketplace for five consecutive trading periods, with a fresh supply of cash or commodity provided at the start of each period.

We implement the CDA marketplace and simulation as described in detail by Tesauro and Das (2001), except for the details of choosing buyer/seller roles and limit prices, as described here. At the start of the game, half of the agents are randomly chosen to be buyers, and the remainder are sellers. Agents are given a list of ten limit prices (seller costs or buyer values) generated from a known random distribution. This distribution uses fixed parameters to generate lower and upper bounds on the limit prices from a uniform distribution, and then generates the limit prices using a uniform distribution between these two bounds. For each run of the game, we randomly select the integer lower bound b of all buyers prices uniformly from $[61,160]$ and the upper bound from $[b+60,b+209]$. We compute the bounds similarly for sellers. The payoff for trading a unit i is $s_i = x_i - l_i$ for sellers and $s_i = l_i - x_i$ for buyers, where x_i is the trade price and l_i is the unit's limit price. The total payoff obtained by an agent in the CDA Game is $\sum_i s_i$. If the efficient number of trades (i.e. the number that maximizes value summed over all agents) at these limit prices is less than 10 or more than 90, we recalculate the bounds and limit prices.

We formulate a normal-form CDA Game of heuristic strategies, abstracting the underlying game of continuous bidding. Each agent chooses a strategy from a set of three alternatives at the start of the game, and does not change during the game. The "Zero-Intelligence Plus" (ZIP) strategy we use (Tesauro & Das 2001) is a modified version of that studied by Cliff and Bruten (1997). ZIP initially bids to obtain a high surplus value (or profit) and subsequently adjusts its bid price price towards the price of any observed trades, or in the direction of improvement when no trades have occurred after a period of time. The Gjerstad-Dickhaut (GD) strategy we use (Tesauro & Das 2001), is a modified version of the original (Gjerstad & Dickhaut 1998). GD calculates a heuristic "belief" function based on the history of recent market activity, and places bids to maximize the expected payoff, given the belief function. The Kaplan strategy (Rust, Miller, & Palmer 1993) withholds bids until the bid/ask spread decreases to a sufficiently small amount or the end of a period is near.

Size	Label	p(ZIP)	p(GD)	p(Kaplan)	Payoff
14	-	0.420	0.000	0.580	0.967
20	**A**	0.439	0.000	0.561	0.972
	B	0.102	0.542	0.356	0.991
	C	0.000	0.690	0.310	0.989

Table 2: The symmetric Nash mixed-strategy equilibria for the CDA Game with 14 and 20 agents. Each row is an equilibrium, showing the probabilities of choosing the high-level strategies (ZIP, GD, and Kaplan), and the expected equilibrium payoff. The labels of the equilibria correspond to those shown in Figure 2(a).

We compute the heuristic-payoff table by averaging the results from 2500 simulations for each profile of strategies.

Analysis of the CDA Game Previous studies show that the choice of strategy in the CDA Game from amongst the alternatives of ZIP, GD and Kaplan is an interesting problem without an obvious solution. Kaplan was the winner of the Santa Fe Double Auction Tournament (Rust, Miller, & Palmer 1993). However, the Kaplan strategy does not perform well against itself, and must be parasitic on the intelligent bidding behavior of other agent strategies to obtain decent profits. A recent analysis (Tesauro & Das 2001) shows that various published bidding strategies all give good empirical performance and that none is dominant. The homogeneous pure-strategy populations are unstable to defection: all-ZIP and all-GD can be invaded by Kaplan, and all-Kaplan can be invaded by ZIP or GD. Hence the Nash equilibria are difficult to compute by inspection or other simple means.

We applied our solution method to the CDA Game with various numbers of agents. We show the equilibria for 14 and 20 agents in Table 2. CDA Games with 6, 12, and 14 agents each result in only one Nash equilibrium with similar mixed-strategy vectors p^* which assign zero probability to choosing the GD strategy. The results were qualitatively different for CDA Games with larger populations. For 16, 18, and 20 agent games we found three equilibria, very similar for each number of agents. One of these equilibria matched the small-population equilibrium, and there are two additional equilibria, one involving only GD and Kaplan, and one using all three strategies.

We plot the replicator dynamics for the 20-agent CDA Game in Figure 2(a). The strategy space is represented by a two-dimensional unit simplex with vertices corresponding to the pure strategies $p=(1,0,0)$ (all ZIP), $p=(0,1,0)$ (all GD), and $p=(0,0,1)$ (all Kaplan). The points labeled **A**, **B**, and **C** are the Nash equilibria shown in Table 2. **A** and **C** are both attractors, while **B** is a saddle point, hence only **A** and **C** are realistic outcomes. We also note that **A** has a much larger basin of attraction than **C**. If the initial p is chosen randomly, the population mix is most likely to terminate at **A** even though **C** has higher population payoff. An

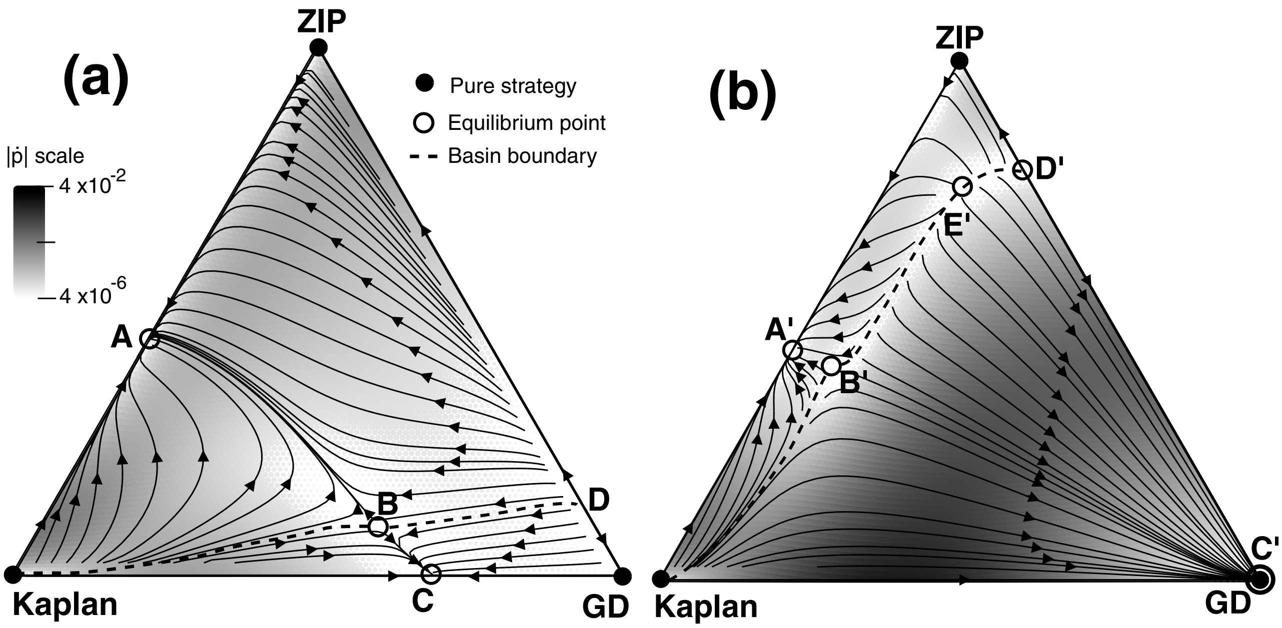

Figure 2: (a) Replicator dynamics for the CDA Game. Open circles are Nash equilibria with labels corresponding to those in Table 2. Other notations are similar to those in Figure 1. (b) Replicator dynamics for the CDA Game with perturbed payoffs, in which 5% of the ZIP and Kaplan agent payoffs was shifted to GD agents.

additional point **D** is shown which is an equilibrium of ZIP and GD in the two-strategy case, but which has incentive to defect to Kaplan in the full three-strategy game.

We also observed the the replicator dynamics for the 14-agent CDA Game. There we found that the single equilibrium was an stable attractor to which all flows converged.

The gray shading in Figure 2(a), which denotes $|\dot{p}|$, indicates that the population changes much more rapidly near the lower-left corner (mostly Kaplan) than near the lower right corner (mostly GD). This shows that there is only a slight incentive to deviate to Kaplan in an all-GD population, and a much larger incentive to switch to ZIP or GD in an all-Kaplan population. Note that $|\dot{p}|$ can vary by up to four orders of magnitude. In particular, the magnitude of the flows leading away from **B** are much smaller than the average $|\dot{p}|$ in the simplex. Hence, although **B** is an unstable equilibrium, the population could actually spend a relatively long time in the region of **B**. The change in this region could be so slow as to appear stable if the base time scale of decision making is not sufficiently fast.

We studied the sensitivity of the population dynamics for the CDA Game shown in Figure 2(b) by simultaneously improving $\sigma^+ = \text{GD}$, relative to $\sigma^- = \text{ZIP}$ and $\sigma^- = \text{Kaplan}$, by $\alpha = .05$. The replicator dynamics of the perturbed CDA Game shows significant change in the topology of the flows, as depicted in Figure 2(b). The right edge of the simplex, corresponding to a mix of GD agents and ZIP agents, is now stable against invasion by Kaplan agents. As a consequence, $\mathbf{D}'$ is now a Nash equilibrium point, and due to the global topology of flows, a new interior equilibrium point occurs at $\mathbf{E}'$. The equilibrium point $\mathbf{C}'$ has moved to the vertex of pure GD. Only $\mathbf{A}'$ and $\mathbf{C}'$ are stable equilibria.

Although pure GD is still not dominant, nor the only attractor, $\mathbf{C}'$ captures much of the simplex in its basin of attraction, making it the most likely attractor in the perturbed CDA Game. Thus, if GD could be improved to capture an extra 5% of other agents' surpluses, it would likely be widely adopted in the population. Moreover, although the payoffs are actually not perturbed at $\mathbf{C}'$ (because it is a pure strategy), we measured that the payoff there is higher than the other perturbed and non-perturbed equilibrium points. We also searched the range of improvements for GD and found that an equilibrium containing GD captures most of the simplex in its basin of attraction when $\alpha > 0.0075$. In short, GD would start becoming a strong strategy with as little as a 0.75% improvement, and would become nearly dominant with a 5% improvement.

Discussion

We have proposed an approach for analyzing heuristic strategies for complex games with repeated interactions. Our methodology treats the heuristic strategies, rather than the component atomic actions, as primitive, and computes expected payoffs to agents as a function of the joint strategy space. With our approach, we can draw useful and general conclusions for games that defy analysis at the level of atomic actions.

We have shown how to apply our methodology to two games whose complexity has thus far defied game-theoretic analysis at the level of atomic actions. For each, we found multiple Nash equilibria. To address the issue of how a particular equilibrium may be realized, we computed the dynamics of a population in terms of the change of proportional shares of strategies. We argued that dynamically unstable equilibria will not be realized, and that the attractor equilibrium with the largest basin of attraction was the most likely to be played in steady-state, while noting the effects of time scale on convergence. We also examined perturbations of the expected payoffs to identify how modest improvements to one strategy could significantly change the dynamic properties and the set of equilibria. For each application we discovered interesting and surprising results not apparent from the simpler analyses commonly applied to heuristic strategies.

Our approach is more principled and complete than the feasible methods commonly applied to the analysis of complex, strategic interactions. Still, more work is needed to provide a full understanding of these interactions. While we have touched on it with perturbation analysis, a deeper understanding of how to design and improve the strategies themselves requires a bridging of action-level and heuristic-level analysis. We have made some preliminary progress along these lines in the CDA game by a direct study of the pure payoff table. There we found that studying the region where a strategy fairs most poorly against others, combined with a deep understanding of the strategies themselves, can provide inspiration for strategy improvements.

Our general approach would also be improved with advances in the specific techniques employed. New learning algorithms should provide improved equilibrium convergence in small populations. As mentioned above, we are exploring techniques to minimize the amount of payoff table computation necessary to accurately determine equilibria. We hope this will make our methodology feasible for analysis of the top TAC strategies. Additionally, we could also perform sensitivity analysis for those games requiring expensive simulation.

Our modeling assumptions and the computational techniques we employ give us the ability to analyze relatively large numbers of agents. This in turn allows us to observe qualitatively different behaviors as the number of agents grows. In some cases it may be reasonable to extrapolate our results to even larger numbers of agents, beyond our ability to directly compute.

The most important computational limitation of our methodology is an exponential dependence on the number of high-level strategies. This would seem to limit its applicability to real-world domains where there are potentially many heuristic strategies. This apparent limitation may be surmountable if the numerous heuristic strategies can be placed into a small number of broad functional groups, and if variations within a group are not as important as the total population fraction within each group. For example, a reasonable approximation for financial markets may be to classify the available trading strategies as either "buy-and-hold," "fundamental," or "technical" strategies, and then carry out our analysis. To allow for algorithmic variations within each group, we could randomly choose between variations for each simulation when computing the payoff table.

Acknowledgments

We thank David Parkes for his helpful comments.

References

Axelrod, R. 1997. Evolving new strategies: The evolution of strategies in the iterated prisoner's dilemma. In *The Complexity of Cooperation*. Princeton University Press.

Borgers, T., and Sarin, R. 1997. Learning through reinforcement and replicator dynamics. *Journal of Economic Theory* 77:1–14.

Cliff, D., and Bruten, J. 1997. Minimal-intelligence agents for bargaining behaviors in market-based environments. Technical Report HPL-97-91, Hewlett Packard Labs.

Foster, D., and Vohra, R. 1997. Regret in the on-line decision problem. *Games and Economic Behavior* 21:40–55.

Freund, Y., and Schapire, R. E. 1996. Game theory, on-line prdiction and boosting. In *Ninght Annual Conference on Computational Learning Theory*.

Fudenberg, D., and Kreps, D. M. 1993. Learning mixed equilibria. *Games and Economic Behavior* 5:320–367.

Fudenberg, D., and Levine, K. K. 1993. Steady state learning and Nash equilibrium. *Econometrica* 61(3):547–573.

Gjerstad, S., and Dickhaut, J. 1998. Price formation in double auctions. *Games and Economic Behavior* 22:1–29.

Greenwald, A., and Kephart, J. 1999. Shopbots and pricebots. In *Proceedings of Sixteenth International Joint Conference on Artificial Intelligence*, volume 1, 506–511.

Greenwald, A., and Kephart, J. O. 2001. Probabilistic pricebots. In *Fifth Internationl Conference on Autonomous Agents*, 560–567.

Greenwald, A., and Stone, P. 2001. Autonomous bidding agents in the trading agent competition. *IEEE Internet Computing* 5(2):52–60.

Greenwald, A.; Kephart, J.; and Tesauro, G. 1999. Strategic pricebot dynamics. In *Proceedings of First ACM Conference on E-Commerce*, 58–67.

Hu, J., and Wellman, M. P. 1998. Multiagent reinforcement learning: theoretical framework and an algorithm. In *Proceedings of the Fifteenth International Conference on Machine Learning*, 242–250. Morgan Kaufmann.

Jennings, N. R., and Wooldridge, M. 2002. Agent-oriented software engineering. In Bradshaw, J., ed., *Handbook of Agent Technology*. AAAI/MIT Press.

Jordan, J. S. 1993. Three problems in learning mixed-strategy Nash equilibria. *Games and Economic Behavior* 5:368–386.

Littman, M. 1994. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the Eleventh International Conference on Machine Learning*, 157–163. Morgan Kaufmann.

McKelvey, R. D., and McLennan, A. 1996. Computation of equilibria in finite games. In *Handbook of Computational Economics*, volume 1. Elsevier Science B. V.

McKelvey, R. D.; McLennan, A.; and Turocy, T. 2000. *Gambit Command Language: Version 0.96.3*.

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

Press, W.; Teukolsky, S.; Vetterling, W.; and Flannery, B. 1992. *Numerical recipes in C*. Cambridge University Press.

Rust, J.; Miller, J.; and Palmer, R. 1993. Behavior of trading automata in a computerized double auction market. In Friedman, D., and Rust, J., eds., *The Double Auction Market: Institutions, Theories, and Evidence*. Addison-Wesley.

Sato, Y.; Akiyama, E.; and Farmer, J. D. 2001. Chaos in learning a simple two person game. Technical Report 01-09-049, Santa Fe Institute.

Singh, S.; Kearns, M.; and Mansour, Y. 2000. Nash convergence of gradient dynamics in general-sum games. In *Proceedings of UAI-2000*, 541–548. Morgan Kaufman.

Smith, V. L. 1982. Microeconomic systems as an experimental science. *American Economic Review* 72:923–955.

Tesauro, G., and Das, R. 2001. High-performance bidding agents for the continuous double auction. In *Third ACM Conference on Electronic Commerce*, 206–209.

Watkins, C. 1989. *Learning from Delayed Rewards*. Ph.D. Dissertation, Cambridge University, Cambridge.

Weibull, J. W. 1995. *Evolutionary Game Theory*. The MIT Press.

Wellman, M. P.; Wurman, P. R.; O'Malley, K.; Bangera, R.; Lin, S.; Reeves, D.; and Walsh, W. E. 2001. Designing the market game for a trading agent competition. *IEEE Internet Computing* 5(2):43–51.