Searching for Approximate Equilibria in Empirical Games

Patrick R. Jordan, Yevgeniy Vorobeychik, and Michael P. Wellman
University of Michigan
Computer Science & Engineering
Ann Arbor, MI 48109-2121 USA
{prjordan,yvorobey,wellman}@umich.edu

ABSTRACT

When exploring a game over a large strategy space, it may not be feasible or cost-effective to evaluate the payoff of every relevant strategy profile. For example, determining a profile payoff for a procedurally defined game may require Monte Carlo simulation or other costly computation. Analyzing such games poses a *search problem*, with the goal of identifying equilibrium profiles by evaluating payoffs of candidate solutions and potential deviations from those candidates. We propose two algorithms, applicable to distinct models of the search process. In the *revealed-payoff* model, each search step determines the exact payoff for a designated pure-strategy profile. In the *noisy-payoff* model, a step draws a stochastic sample corresponding to such a payoff. We compare our algorithms to previous proposals from the literature for these two models, and demonstrate performance advantages.

Categories and Subject Descriptors

I.2.8 [**Problem Solving, Control Methods, and Search**]: Heuristic methods; I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Economics, Experimentation

Keywords

Empirical game, approximate equilibria, heuristic search

1. INTRODUCTION

In attempting to understand agent interactions in multiagent systems, researchers often appeal to game-theoretic solution concepts to characterize the strategic stability of hypothetical outcomes. Unfortunately, the strategy space of the game or interaction being modeled is often so complex to render infeasible exact game-theoretic modeling and analysis. One common compromise is to consider stylized versions of the game that are amenable to computational analysis, at the expense of fidelity. One alternative pursued by experimental AI researchers in recent years is to estimate games through simulation and sampling [3, 9, 17], an approach that has been termed *empirical game-theoretic analysis* [18].

Cite as: Searching for Approximate Equilibria in Empirical Games, Patrick R. Jordan, Yevgeniy Vorobeychik, and Michael P. Wellman, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16.,2008,Estoril,Portugal,pp.1063-1070.

In empirical game modeling, the outcome of a joint strategy, or *profile*, is estimated by repeatedly sampling the game. These samples can be generated by a game simulator or other model describing the game. Such an approach is quite general, but incurs an estimation cost in proportion to the size of the profile space, which is exponential in the number of players and the number of strategies available per player. For many games of interest, the strategy set is extremely large or even infinite. Thus, in practice we cannot explore the space exhaustively, but instead focus on profiles that are most promising as solutions or otherwise pivotal in game-theoretic analysis.

Figure 1 presents an overview of the empirical game-theoretic analysis framework we adopt in this study. In place of an analytic description of the mapping from strategy profiles to payoffs, a simulator generates a set of sample observations. This set of samples, or the model estimated or inferred from them, constitutes the *empirical game*. Strategic reasoning about the empirical game description can support the design of agents to play the game, or guide the design of multiagent interaction mechanisms that induce such games. For both of these design problems, we typically seek to characterize *solutions* of the game, for example by identifying exact or approximate Nash equilibria. Finding these solutions thus constitutes a central search problem for empirical strategy design and empirical mechanism design.

Previous research has explored directed sampling of profiles, by using value of information estimates [17] or interleaving sampling and equilibrium calculations [10]. Both techniques require at least a small number of samples to be generated for every profile in the full joint strategy space. Since it may be possible to establish that a particular profile is an equilibrium or near-equilibrium without considering all profiles, a search approach can potentially relax this requirement. This was part of the motivation for Sureka and Wurman [14], who proposed an algorithm based on tabu best-response search to search for pure-strategy Nash equilibria within the profile space.

This latter algorithm is applicable in a *revealed-payoff* search model, where each search step determines the exact payoff for a designated pure-strategy profile. In contrast, the directed sampling methods described above assume a *noisy-payoff* model, where the basic search step corresponds to drawing a sample from an underlying distribution of payoffs.

In this paper, we propose new algorithms for both search models, and compare them to the previous approaches from the literature. For the revealed-payoff model, we develop an approach based on minimum-regret-first search, and find that this algorithm is comparable on one measure of search efficiency and superior on another to tabu best-response [14]. Next, we describe a repeated sampling algorithm termed information-gain search, applicable to

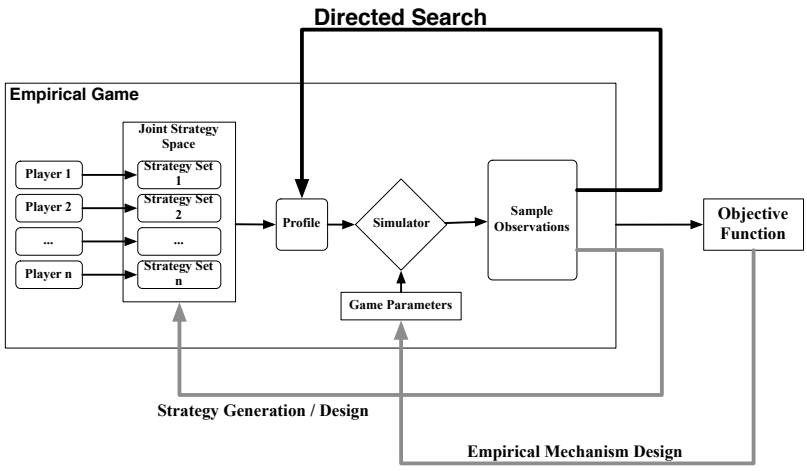

Figure 1: Overview of empirical game-theoretic analysis where directed search is used to reduce the number of observations sampled in confirming profiles with low ϵ. These profiles are extensively used in strategy and mechanism design.

the noisy-payoff model. In experimental comparison, we find that information-gain search outperformed the literature benchmark [17] for both low and high degrees of noise.

2. NOTATION

This section describes the formal notation and experimental measures of success we use to compare algorithms.

DEFINITION 2.1 (NORMAL FORM GAME). $\Gamma = \langle I, \{S_i\}, \{u_i(s)\}\rangle$ *is a* normal form game*, with I the set of players, S_i the set of strategies available to player i, and $u_i : \times_{j=1}^{|I|} S_j \rightarrow \mathbb{R}$ the utility function for player i mapping the joint strategy s to the real-valued payoff received by player i when s is played.*

In the case where the payoffs are random variables, $u_i(s)$ represents the expected von-Neumann-Morgenstern utility for playing joint strategy s. We refer to the joint strategy set $\times_{j=1}^{|I|} S_j$ for a particular game Γ as S. The term *profile* is interchangeable with *joint strategy*.

Our search and analysis focuses on *pure* profiles, where strategies are selected deterministically.[1] Each profile is associated with the set of neighboring profiles that can be reached through a unilateral deviation by one player.

DEFINITION 2.2 (UNILATERAL DEVIATION SET). *The* unilateral deviation set *for player i and profile $s \in S$ is*

$$\mathcal{D}_i(s) = \{(\hat{s}_i, s_{-i}) : \hat{s}_i \in S_i - \{s_i\}\},$$

and the corresponding set for an unspecified player is

$$\mathcal{D}(s) = \bigcup_{i \in I} \mathcal{D}_i(s).$$

For a given player i, the *best-response correspondence* for a given profile s is the set of strategies which yield the maximum payoff holding the other players' strategies constant.

[1]Many but not all of the methods we describe can be straightforwardly extended to admit mixed strategies, where players choose actions probabilistically. We maintain exclusive consideration of pure strategies here for simplicity, deferring comprehensive coverage of the mixed-strategy case to future work.

DEFINITION 2.3 (BEST RESPONSE). *For some joint strategy $s \in S$, the player i best-response correspondence is*

$$\mathcal{B}_i(s) = \underset{\hat{s} \in \mathcal{D}_i(s) \cup \{s\}}{\arg\max} u_i(\hat{s})$$

The overall best-response correspondence is then given by

$$\mathcal{B}(s) = \prod_{i \in I} \mathcal{B}_i(s)$$

The *best-response dynamic* is the result of iteratively applying the best-response correspondence. A *pure-strategy Nash equilibrium* (PSNE) is a fixed point in this process, that is, $s \in \mathcal{B}(s)$.

The goal of game-theoretic search is to identify profiles that are strategically *stable*, with Nash equilibrium representing perfect stability. To evaluate relative stability of a profile, we measure its *regret*, the maximum gain from deviation available to any player.

DEFINITION 2.4 (REGRET). *The regret of strategy profile $s \in S$, $\epsilon(s)$, is the maximum gain from deviation from s by any player. Formally,*

$$\epsilon(s) = \max_{i \in I,\ \hat{s} \in \mathcal{D}_i(s) \cup \{s\}} u_i(\hat{s}) - u_i(s).$$

The regret of a Nash equilibrium is zero. More generally, we say that profile s is an $\epsilon(s)$-Nash equilibrium, which means it approximates equilibrium at the level $\epsilon(s)$. Approximate equilibria (low-regret profiles) may be of interest in general, and are especially salient when searching among pure profiles for games that may not exhibit pure-strategy Nash equilibria.

Finally, we present our notion of an empirical game, formally defined in terms of the evidence we have for profile payoffs.

DEFINITION 2.5 (EMPIRICAL GAME). *Let $\Gamma = \langle I, \{S_i\}, \{u_i(s)\}\rangle$ be a game, and θ a set of evaluations for the payoff function u. Then $\mathcal{E}(\Gamma, \theta) = \langle I, \{S_i\}, \theta\rangle$ constitutes an* empirical game *for Γ.*

Under the revealed-payoff model, each evaluation in θ gives the value of the payoff function for some profile. Under noisy payoffs, each evaluation in θ gives a noisy sample of the true payoff for some profile. We use the notation $\mathcal{E}.s^k$ to denote the resulting empirical game after k more evaluations of profile s.

3. PROBLEM STATEMENT

Our starting point is a game (the true, or base game) in which the payoff function is specified by a simulator. For any available number of calls to the simulator, our objective is to obtain the highest-quality solution we can to this game. This problem can be divided into two interdependent parts. The first part is control of sampling effort, to produce an empirical game given the available number of simulator calls. The second is to generate from the empirical game a candidate solution to the base game. Naturally, the control algorithm may make use of intermediate generated solution profiles to effectively guide the collection of further samples. And since performance will be measured based on the solution ultimately produced, the control objective is shaped by the criterion for generating solutions.

We measure quality of the generated solution profile by regret—the maximum gain to deviation from this solution in the *base* game. To measure performance of our search algorithms, we generate a solution on each iteration, calculate its regret with respect to the base game, and repeat this for many randomly generated problem instances. This allows us to characterize solution quality as a function of search time, for example to evaluate the average time to achieve various levels of quality.

For search under the revealed-payoff model, we can draw a further distinction based on whether the regret is known in the empirical game, or merely bounded based on deviations evaluated thus far. We say that a profile s is *confirmed* in the empirical game if all deviations $s' \in \mathcal{D}(s)$ have been evaluated. In some cases, we require that search algorithms return only confirmed solutions. Since the regret of confirmed profiles is known, the best solution to return is obviously the one with minimum regret.

4. SEARCH METHODS FOR REVEALED PAYOFFS

Our first setting involves games in which the payoff simulator returns an exact payoff when queried with a particular profile. We know of only two search algorithms for this model studied previously in the literature: one based on TABU best response (or simply, TABU) [14], and another applying regret bounds in a minimum-regret-first search (MRFS), employed by Vorobeychik et al. [15]. Below, we describe the two methods and, in the section that follows, evaluate them experimentally on several classes of randomly generated games.

4.1 TABU Best-Response Search

The TABU best-response algorithm [14] begins by selecting an arbitrary profile as *active*. Subsequently, each iteration involves (a) selecting a "deviant" player i, (b) finding a most profitable deviation for i from the current active profile s, (c) selecting the profile $s' = (t_i, s_{-i})$ as the next active profile, where t_i is the best response of i to s, and (d) adding either s_i in the *attribute based memory* version or s in the *explicit memory* version of the algorithm to the *tabu* list, L. When the attribute based memory version of the algorithm is used, deviation options for player i are restricted to strategies not in the tabu list. When the explicit memory version of the algorithm is used, deviation options for player i are restricted to strategies not yielding profiles in the tabu list. The process terminates once the algorithm selects a PSNE as its active profile.

The original experiments by Sureka and Wurman evaluate performance based on the number of search steps required to find a PSNE. Since the experimenters know the base game and therefore its equilibria, they can simply terminate search when one of the known PSNE becomes the active profile.

In practice, when searching an unknown game, the algorithm cannot generally determine that an equilibrium profile is actually such when it first becomes active.[2] Thus, we also consider performance measures that require generated solutions to be confirmed, as discussed above. Addressing this confirmation requirement required that we modify TABU to seek confirmation rather than move always to best response. In the modified version, instead of immediately placing the active profile on the tabu list L and branching to the best response, we do so only if the best response strictly increases the player's payoff; otherwise we keep the active profile unchanged. With this modification, we can confirm an active profile upon iterating through all the players. However, it now becomes possible in the explicit memory version of the algorithm (the version used in the experiments below) that we visit a profile for which all neighbors are in the tabu list. In this case we allow the player to deviate to the best response if that best response gives a higher payoff than the current profile. Pseudo-code for the tabu best-response algorithm used in our experiments is presented below.

TABU-BEST-RESPONSE-SEARCH

$L \leftarrow \emptyset$
Select initial profile at random
while termination criteria not satisfied
  do
    $i \leftarrow$ next player
    if $\mathcal{D}_i(s) \subseteq L$
      then $s \leftarrow$ player i's best response to s
      else if s has an improving deviation in $\mathcal{D}_i(s) \setminus L$
        then Push s onto L
        $s \leftarrow$ player i's best response to s not in L

4.2 Minimum-Regret-First Search

The idea of minimum-regret-first search (MRFS) is to expand a search tree by exploring the fringe node that is best according to some priority measure. In our setting, the objective is to find a profile minimizing the maximal gain from deviation. Therefore we adopt as our priority measure a lower bound, $\hat{\epsilon}(s)$, on the possible gain to deviation from profile s, which is just the greatest gain among deviations from s that have been evaluated. The pseudo-code below describes the MRFS procedure.

MINIMUM-REGRET-FIRST-SEARCH

Select initial profile at random
while Queue is not empty
  do
    Select lowest $\hat{\epsilon}(s)$ profile s from queue
    if s is confirmed
      then Remove it from queue and assign $\epsilon(s) = \hat{\epsilon}(s)$
      else $\bar{s} \leftarrow$ SELECT-DEVIATION(s)
        Insert $\bar{s}$ into queue if previously unevaluated
        Update $\hat{\epsilon}(\hat{s})$ for $\hat{s} \in \{\bar{s}\} \cup \mathcal{D}(\bar{s})$ in the queue

---

The subroutine SELECT-DEVIATION(s) returns some deviation from s which has yet to be sampled. In this selection we try to predict which unevaluated deviation from s is likely to give the largest gain from deviation. While the efficacy of the deviation selection heuristic depends on the game class, we have empirically found one heuristic that works well in a variety of cases. Specifically,

[2]Moreover, in general we cannot assume that a PSNE even exists for the base game. We can relax the criterion to allow approximate equilibria, though we typically do not know *a priori* the regret of the best pure-strategy approximate solution.

our method tracks deviations by player index and target strategy, and selects the unevaluated deviation from the current profile that has most frequently produced an improvement in the search history thus far.

5. EVALUATION OF SEARCH METHODS FOR REVEALED PAYOFFS

Our experiments employ games of various classes generated by GAMUT [8]. When applicable, we select game instances similar in size to those used by Sureka and Wurman [14]. Initially we experiment with a game class used in their prior study to establish a baseline for algorithm comparison. We then proceed to investigate a game class whose structure is known to be exploited by a best-response dynamic, so that we may compare the algorithms in an environment expected to be favorable to TABU. Results for other game classes were qualitatively similar, reported in a prior workshop version of our revealed-payoff algorithm study [5].

5.1 Uniform Random Games

Our first class of games has payoffs that are uniformly and independently distributed in the range [-100,100]. The game class is denoted URG($|I|, |S_i|$), where $|I|$ is the number of players and $|S_i|$ is the strategy set size of each player. We compare MRFS and TABU on two sizes of games: the smaller URG(5,5) and the larger URG(5,10). To construct the data sets for comparing the algorithms we generated 20 games in each class, and checked which instances possess a PSNE. For each instance we ran each algorithm 100 times, with randomly selected starting profiles on each run.

Our first comparison measures the number of evaluation steps (expressed in terms of percentage of profile space) required to confirm an equilibrium. For this measure, we necessarily limited attention to those games possessing a PSNE. The results of this comparison are shown in Table 1.

	URG(5,5)		URG(5,10)	
	Mean (%)	Median (%)	Mean (%)	Median (%)
MRFS	53.42	52.12	37.10	31.41
TABU	52.25	49.28	41.79	34.75
p value	0.18		3.5e-05	

Table 1: Percentage of profile space explored to confirm a PSNE, among URGs with at least one PSNE.

Our analysis of URG(5,5) included 12 games which contained at least one PSNE. Seeds 0, 1, 3, 4, 8, and 15 contained one PSNE; seeds 5, 13, 16, 17, and 18 contained two; and seed 20 contained three. The performance of MRFS and TABU varied drastically according to the individual game. For instance, in seeds 4, 8, and 15, TABU rarely succeeded in confirming the solution.[3] Similarly in seed 3, MRFS on average requires nearly all the search space to be evaluated. It should be noted that although the equilibrium was not confirmed until near the last iteration, many near-equilibrium profiles were confirmed much earlier.

Our analysis of URG(5,10) also included 12 games which contained at least one PSNE. Algorithm performance differences are statistically significant in the large game. In these larger games the average performance of MRFS and TABU improves from approximately 50% of the space searched to the mid-30% range.

Figure 2 shows the minimum confirmed ϵ as a function of the space explored, which is our second performance measure. In many practical settings it may be that near equilibrium profiles are just as useful as PSNE. Therefore in those cases we consider the second measure more appropriate. Notice that MRFS confirms low ϵ profiles much earlier than TABU, which is the desired result.

5.2 Congestion Games

The second comparison we present is an experiment using *congestion games*. The GAMUT [8] user documentation describes the class as follows:

> In the congestion game, each player chooses a subset from the set of all facilities. Each player then receives a payoff which is the sum of payoff functions for each facility in the chosen subset. Each payoff function depends only on the number of other players who have chosen the facility.

A convenient feature of congestion games is that they possess a potential function [11]. As a consequence, they exhibit two key properties for our purposes: they possess PSNE, and best-response learning processes converge to this equilibrium [7]. This latter property suggests that the TABU best-response search algorithm should be effective.

We compare MRFS and TABU on four-player four-facility congestion games. Congestion(4,4) has $65,536$ distinct profiles since each player chooses a subset of the 4 facilities to play. Exploiting player symmetry would reduce this to 3876 distinct profiles, though in our experiments the algorithms do not do this. Exploiting symmetry would of course have only improved performance for the fixed game size. The results of the congestion game comparison are shown in Table 2. Twenty games were generated for experimentation using GAMUT. As expected, these games were extremely easy for TABU, which needed to search only a tenth of one percent of the profile space on average to confirm a PSNE. They were quite easy for MRFS as well, although this algorithm required 0.15 of one percent. The differences are statistically significant, but practically negligible given the miniscule amount of search required.

	MRFS	Tabu
Mean (%)	0.15	0.10
Median (%)	0.15	0.10
p value	$<$ 2.2e-16	

Table 2: Congestion game (4,4). Percentage of profile space explored to confirm a PSNE.

6. SEARCH METHODS FOR NOISY PAYOFFS

When payoff realizations are noisy, it is clear that the MRFS and TABU best-response algorithms are inadequate, since these do not consider how to allocate samples across evaluated profiles. Nevertheless, we can apply them to the problem in a modified form, interpreting an evaluation step as a decision to draw k payoff samples for the target profile. Clearly, as k increases, so does reliability of the answers. On the other hand, increasing k reduces the number of profiles that can be explored with a given number of samples.

[3]Since TABU is not guaranteed to confirm an existing solution a timeout was placed on the number of iterations equal to the size of the strategy space. If TABU exceeded the timeout it was credited for finding the solution in the greatest possible number of steps required by MRFS.

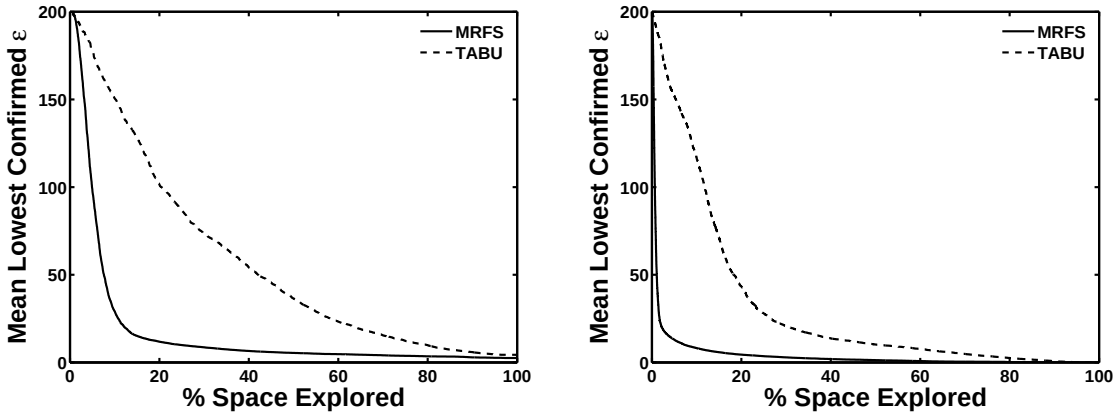

Figure 2: Mean lowest confirmed regret (ϵ) for URG(5,5) on the left and URG(5,10) on the right.

An analyst can, perhaps, guess a reasonable value for k for a particular problem, using higher k when more noise is present. Such a solution is unsatisfactory for two reasons. First, it seems a waste to sample each profile an equal number of times, since some profiles will show to be hopeless as solution candidates after only a few samples. Second, we would like to develop an algorithm that can automatically adjust sampling allocation appropriately for a given problem, rather than involve the analyst in the process.

We are aware of two previous approaches to the problem of sample allocation in games with noisy payoffs, one by Walsh et al. [17] and another by Reeves et al. [10]. Walsh et al. introduced a search algorithm founded on the principles of metareasoning [13], using an approximate regret function to determine the value of choosing a specific profile to sample. Reeves et al. proposed a method of search by which profiles are selected to be sampled according to the estimated probability mass placed on the specific profile in a sample equilibrium. Below we introduce another approach which is based on information gain as measured by the Kullback-Leibler divergence criterion [6].

6.1 ECVI (Walsh et al.)

First, we discuss the approach to guided search in noisy games introduced by Walsh et al. [17]. Below we overload s to denote a profile $s \in S$ as well as the action of sampling it and let S^T be the space of sequences of sampling actions of length T. We also use the notation $\mathcal{E}.s$ to indicate an empirical game which results by sampling a profile s and adding the resulting sample payoff to the current data set in $\mathcal{E}$. Let $x(\mathcal{E})$ be some decision model based on the information encompassed within the current empirical game and let $\psi(s)$ be the *error model* for selecting a profile s. We can define the *expected value of information* (EVI) from sampling a particular profile s under the current information state $\mathcal{E}$ to be

$$EVI(s|\mathcal{E}) = \mathbb{E}_{\mathcal{E}.s|\mathcal{E}}\left[\, \psi_{\mathcal{E}.s}(x(\mathcal{E})) - \psi_{\mathcal{E}.s}(x(\mathcal{E}.s)) \,\right]. \quad (1)$$

Walsh et al. propose two algorithms. The first of these is EVI as defined in Equation 1, in which $x(\mathcal{E})$ selects an arbitrary, possibly mixed-strategy, Nash equilibrium in the empirical game $\mathcal{E}$. Additionally, their error model is *cumulative regret*, defined to be

$$\psi(s) = \sum_{i \in I,\ \hat{s} \in \mathcal{D}_i(s)} \max(0,\ u_i(\hat{s}) - u_i(s)).$$

Finally, they develop a particular model of future information which uses distributional estimates for the payoffs $u_i(s)$ of the form

$$\mu_{\mathcal{E}.s^\infty}(u_i(s)) = \mu_{\mathcal{E}}(u_i(s)) \quad \text{and} \quad \sigma_{\mathcal{E}.s^\infty}(u_i(s)) = \frac{\sigma_{\mathcal{E}(u_i(s))}}{n_{\mathcal{E}}(s) + |s^\infty|}$$

where s^∞ is some large repeated sampling sequence of s, $n_{\mathcal{E}}(s)$ is the number of samples of profile s already in $\mathcal{E}$, and $\sigma_{\mathcal{E}(u_i(s))}$ is the sample variance of $u_i(s)$.

When calculating the expectation of $\psi_{\mathcal{E}.s}(x(\mathcal{E}.s))$, Walsh et al. generate Monte Carlo samples which each define a future error function $\psi_{\mathcal{E}.s}(\cdot)$. For each of these samples, a new set of Monte Carlo samples is generated for the decision process $x(\mathcal{E}.s)$. Walsh et al. found that this EVI approach is computationally infeasible for large games, which motivated their development of a second algorithm termed *expected confirmational value of information* (ECVI). Whereas in EVI sampling s has positive value in expectation only if it is expected to change (refute) the current equilibrium choice, ECVI gives more value to s if it is likely to *confirm* the current equilibrium $x(\mathcal{E})$. This was done in an attempt to approximate the true regret and decision functions in the base game. Specifically, the authors [17] calculate ECVI as

$$ECVI(s|\mathcal{E}) = \mathbb{E}_{\mathcal{E}.s|\mathcal{E}}\left[\, \psi_{\mathcal{E}}(x(\mathcal{E})) - \psi_{\mathcal{E}.s}(x(\mathcal{E})) \,\right].$$

To help understand the implications of using ECVI, we consider an alternate formulation of the expected error of $\psi_{\mathcal{E}.s}(x(\mathcal{E}.s))$. Instead of generating Monte Carlo samples for $\psi(\cdot)$ and $x(\cdot)$ independently, we use each additional sample for the error and decision function's empirical game $\mathcal{E}.s$. Because a Nash equilibrium will always exist in our finite $\mathcal{E}.s$, we know that one will be returned by the decision process $x(\mathcal{E}.s)$. We know from the definition of the error function that for each Monte Carlo sample, $\psi_{\mathcal{E}.s}(x(\mathcal{E}.s))$ will be zero. Therefore the EVI equation is simplified to

$$EVI(s|\mathcal{E}) = \mathbb{E}_{\mathcal{E}.s|\mathcal{E}}\left[\, \psi_{\mathcal{E}.s}(x(\mathcal{E}))\right]. \quad (2)$$

Notice that in ECVI the left term in the expectation is not a random variable and is constant for all s. Both EVI and ECVI seek to maximize the value of their respective information notions. Thusly EVI will chose the s which maximizes the expectation in Equation 2, while ECVI will chose the s which minimizes it.

Intuitively, the left and right terms in the expectation of Equation 1 provide an implicit balance between exploitation and exploration, respectively. Under its entailing assumptions, the exploration component vanished in Equation 2. Notice that EVI will choose a profile s which is the most likely in expectation to refute the current

candidate solution. This is precisely what MRFS attempts in the revealed payoff domain by selecting unobserved deviations from the current candidate solution. Thus Equation 2 also provides a qualitative link between EVI and MRFS.

While our implementation of ECVI adopts the future information model used by Walsh et al. without change, this is not so with their decision and error models. Since we could be dealing with very large games in which computing a sample mixed strategy Nash equilibrium is in general an intractable operation, we restrict their decision model to select a pure strategy Nash if one exists, and a pure profile with smallest regret otherwise. Furthermore, our error model is regret, $\epsilon(s)$ as defined above, rather than cumulative regret that they use.

Like the Walsh et al. error model, $\epsilon(\cdot)$ has a zero at a Nash equilibrium. However, the ϵ regret measure has different properties with regards to non-zero error profiles. Example 1 highlights the difference between the two. The Walsh et al. error model will assign the same value (10) to both profiles. The ϵ error model will assign 1 to the first profile and 10 to the second. We believe the ϵ measure is more representative of agent regret.

EXAMPLE 1. *Consider a profile in symmetric game Γ_1 with 10 different deviations. In each of the deviations the deviating player gains 1.*

Now consider a profile in another symmetric game Γ_2 with 10 different deviations. In one of the deviations the deviating player gains 10. The other deviations give the deviating player a gain of 0.

6.2 Information Gain Approach

In this section we present our core algorithm for intelligent search in games: the information gain algorithm. While past approaches, such as EVI and ECVI above, focus on improving a particular (perhaps arbitrary) Nash equilibrium estimate (in our setting, a pure strategy equilibrium), the information gain approach focuses on improving a model which is based on any model which yields the *distribution* over profiles given an empirical game. Such profile distributions arise, for example, as *belief distributions of play* [16], which are beliefs constructed by an outside observer (e.g., mechanism designer) about the relative likelihood of different profiles arising as a result of actual strategic interaction which is modeled by the game. Belief distributions of play may model players as selecting an arbitrary Nash equilibrium, or may involve more complex beliefs—for example, a probability distribution which assigns higher probability to profiles with lower regret will likely assign positive (albeit often small) probability to every profile in a finite game. Below, we are interested in a particular such belief model which assigns the relatively likelihoods to profiles based on their respective probabilities of having the smallest regret.

Our information gain algorithm is, in principle, straightforward. We begin by presuming that our sampling action will take k samples. With each profile $s \in S$ we compute (or approximate) information gain from sampling this profile k times. We then select the profile which promises the greatest information gain. The core of the approach to computing information gain, based on Kullback-Leibler divergence, is very general in that it can use any prior distribution on profiles obtained based on the current empirical game, $p_s(\mathcal{E})$. Thus, we first develop it for an arbitrary distribution, and then specialize to use the one of particular interest to us in this work.

First, we define the *entropy* of a profile s, $\mathcal{H}(s;\mathcal{E})$:

$$\mathcal{H}(s;\mathcal{E}) = -p_s(\mathcal{E})\log_2 p_s(\mathcal{E}) - (1-p_s(\mathcal{E}))\log_2(1-p_s(\mathcal{E}))$$

The standard definition of *cross entropy* of s, denoted here by $\mathcal{H}(s;\mathcal{E},\hat{\mathcal{E}})$, is then

$$\mathcal{H}(s;\mathcal{E},\hat{\mathcal{E}}) = -p_s(\mathcal{E})\log_2 p_s(\hat{\mathcal{E}}) - (1-p_s(\mathcal{E}))\log_2(1-p_s(\hat{\mathcal{E}}))$$

Based on these, we define the *information gain* for a profile s from taking k additional samples of $\hat{s}$, denoted $\mathcal{G}(s;\mathcal{E},\mathcal{E}.\hat{s}^k)$, to be

$$\mathcal{G}(s;\mathcal{E},\mathcal{E}.\hat{s}^k) = \mathcal{H}(s;\mathcal{E},\mathcal{E}.\hat{s}^k) - \mathcal{H}(s;\mathcal{E})$$

Finally, the aggregate information gain from sampling a profile s a total of k times, denoted $\mathcal{G}(\mathcal{E},\mathcal{E}.s^k)$, is

$$\mathcal{G}(\mathcal{E},\mathcal{E}.s^k) = \sum_{\hat{s}\in D(s)} \mathcal{G}(\hat{s};\mathcal{E},\mathcal{E}.s^k)$$

The information gain so defined is then used as a part of our *Info-Gain-Search* selection algorithm:

INFO-GAIN-SEARCH($\mathcal{E}(\emptyset); k; T$)
  Select initial profile at random s
  $\mathcal{E} \leftarrow k$ samples of s
  while Termination criteria not satisfied
    do
      $s \leftarrow \arg\max_{\hat{s}} \mathbb{E}_{\mathcal{E}.\hat{s}^k|\mathcal{E}}\left[\mathcal{G}(\mathcal{E},\mathcal{E}.\hat{s}^k)\right]$
      $\mathcal{E} \leftarrow \mathcal{E} \cup k$ samples of s
  return $\arg\max_{\hat{s}} p_{\hat{s}}(\mathcal{E})$

In developing our assessment of likely strategic outcomes based on the evidence encompassed by the empirical game, we posit that players are most likely to play a profile with the lowest regret. Since we restrict our search space to pure strategy profiles, such profiles need not constitute Nash equilibria, although often they will (particularly in very large games), and even more often the smallest regret will be indeed quite low to justify our belief. Thus, we define our information gain with respect to the distribution $p_s(\mathcal{E})$ which assigns probabilities to profiles s in proportion to their likelihood of having the smallest regret. We now develop these distributions formally, beginning with the definition of the highest payoff a player i can obtain by deviating from s to another strategic option:

DEFINITION 6.1 (MAXIMUM DEVIATION PAYOFF). *For a given player i and profile s, the maximum deviation payoff is*

$$\delta_i(s) = \max_{\hat{s} \in S_i \setminus s_i} u_i(\hat{s}, s_{-i}).$$

The distribution of $\delta_i(s)$, denoted by $F_{\delta_i(s)}(\delta)$, is the n^{th} order statistic (maximum) over the mean payoffs of the deviations, given by

$$F_{\delta_i(s)}(d) = \prod_{\hat{s} \in S_i \setminus s_i} F_{u_i(\hat{s}, s_{-i})}(d).$$

The distribution of player regret, r, denoted by $F_{\epsilon_i(s)}(r)$ can be obtained by conditioning on the payoff to i from playing s:

$$F_{\epsilon_i(s)}(r) = \int_{\mathbb{R}} F_{\delta_i(s)}(u+r)\cdot dF_{u_i(s)}(u). \qquad (3)$$

We estimate the integral in (3) using Monte Carlo with importance sampling [12]. The distribution of regret for a particular profile, s, is then simply

$$F_{\epsilon(s)}(r) = \prod_{i\in I} F_{\epsilon_i(s)}(r).$$

Now, as the final piece, we can define the actual distribution of minimum regret, that is, we can define, for each profile $s \in S$, the probability that s has minimum regret given the evidence in the empirical game:

$$p_S(\mathcal{E}) = \int_{\mathbb{R}} \left[\prod_{\hat{s} \in S \setminus s} \left(1 - F_{\epsilon(\hat{s})}(r)\right) \right] dF_{\epsilon(s)}(r). \quad (4)$$

To estimate the value of the integral in (4) using Monte Carlo, we have to generate M realizations of the random variable. Each of these M realizations requires computing or estimating the Equation (3) expression $|S| - 1$ times. The latter, as we already mentioned, is also estimated using Monte Carlo by generating N realizations of its respective random variable. Thus, each iteration requires $\mathcal{O}(|S|NM)$ operations, for at total running time of $\mathcal{O}(|S|NM\frac{T}{k})$. Furthermore, we may have to use a very large M to get a reasonable approximation. Consequently, running time of our algorithm quickly becomes impractically long. To keep it somewhat in check, we instead approximate the integral by using point estimates for the mean regret of the remaining profiles:

$$p^*_S(\theta) = F_{\epsilon(s)}(\epsilon^{(1)}_{S \setminus s}), \quad (5)$$

where $\epsilon^{(1)}_{S \setminus s}$ is the lowest regret over all profiles except s calculated using the expected mean payoffs given the empirical game $\mathcal{E}$. The approximation in (5) requires $\mathcal{O}(|S|N)$ calculations in each iteration, for a total running time of $\mathcal{O}(|S|N\frac{T}{k})$.

7. EVALUATION OF SEARCH METHODS FOR NOISY PAYOFFS

Unlike our evaluation of search algorithms for games with revealed payoffs, which used randomly generated games of various classes, we evaluate the approaches for noisy games in a more representative setting. The base game used for this purpose is a Supply-Chain Management game in the Trading Agent Competition (TAC/SCM [1, 2]) with the strategy sets of players comprised of heuristic strategies. The game is modeled as a symmetric normal form game with five heuristic strategies[4], for a total of 35 strategy profiles. The payoffs in the game are estimates based on collected data for every strategy profile of a three-player reduction of the original six-player game, obtained using the hierarchical game reduction technique [19]. We use those estimated payoffs [4] to construct a base game which has structure similar to the true TAC/SCM game. Therefore it is with respect to the approximated TAC/SCM base game that we measure error. This technique, in our application, creates three player-pairs. Each of these pairs is constrained to play the same strategy and the payoff for the pair is the average of the payoffs of each member in the original game. In typical TAC/SCM analysis, sampling is the dominant cost, taking nearly an hour per data point. In our study, we eliminate this cost by simulating additive zero-mean normal Gaussian noise on top of the already sampled base game.

We present the results of two experiments. The first experiment uses Gaussian noise with standard deviation of 3.75 million, which is roughly the order of magnitude of the noise found in TAC/SCM simulations. The second experiment has a larger standard deviation of 10 million. For each of these experiments, we tested four different algorithms. For each experiment and algorithm we generated 100 runs and average the score over runs. The score is the true regret ϵ of the returned profile in the base game as a function of the number of samples.

The first algorithm tested was the MRFS extension to noisy games, labeled MRFS-30 in Figure 3. MRFS-30 samples each profile 30 times and uses the resultant mean as if it were the actual payoff in a revealed payoff game.

Secondly, we tested the IGS and ECVI algorithms. These are repeated sampling algorithms and normally require some initial samples of every profile in the game. Therefore we prefixed the repeated sampling portion of the search with a MRFS search where 3 samples are taken per profile. Each iteration of IGS and ECVI algorithms took 5 samples of each profile per iteration. These algorithms were labeled IGS-MRFS-3 and ECVI-MRFS-3, respectively.

Finally we tested the IGS algorithm with a zero-mean Gaussian prior payoff distribution over profiles. In the small-variance game the standard deviation of the Gaussian prior was taken to be 5 million, whereas in the large-variance game it was 20 million. This algorithm was labeled IGS-WITH-PRIOR.

Figure 3 shows the results of the analysis. In the small variance game we see that MRFS-30 does not perform as well as the other algorithms for most of the sample sizes. Using MRFS-3 to gather initial samples seemed to help substantially for the first 200 samples, after which IGS-WITH-PRIOR caught up with the performance of IGS-MRFS-3. Note that MRFS-3 uses up the first 105 samples. Finally, we note that IGS-MRFS-3 offers a performance improvement over ECVI-MRFS-3.

In the large-variance game we note the surprising result that all of the algorithms outperformed ECVI-MRFS-3, particularly when more samples were taken. MRFS-30 display a particularly strong performance in this game class, essentially on par with IGS-MRFS-3 and IGS-WITH-PRIOR.

8. DISCUSSION

We have investigated the problem of searching for approximate equilibria in games where determining the payoff for particular profiles is costly. We considered two models of payoff evidence:

- revealed-payoff: each search step evaluates a profile, obtaining exact payoff information
- noisy-payoff: each search step produces a sample stochastically generated conditional on actual payoffs

For each model, we experimentally evaluated the known approaches from prior literature—all, as far as we are aware—along with new algorithms and variants, on a range of game instances and game classes.

For the revealed-payoff model, we compared MRFS and TABU on classes of games with or without helpful structure. In all cases, we found that the methods require approximately the same number of search steps on average to confirm a PSNE. MRFS significantly outperforms TABU, however, in terms of its ability to confirm better approximate equilibria earlier, for games that require significant search. Another important attribute of the MRFS algorithm is that it will confirm all available profiles eventually, whereas TABU may not. This is important not only in the case where no PSNE exists, but also when we wish to analyze low-regret profiles when designing a best response.

For the noisy-payoff model, we introduced a new algorithm based on information gain, called IGS, and found that it outperforms the ECVI repeated sampling algorithm, the current benchmark in the

[4]The heuristic strategies are a subset of the agents who participated in the TAC/SCM 2006 tournament and released binary versions of their agent software.

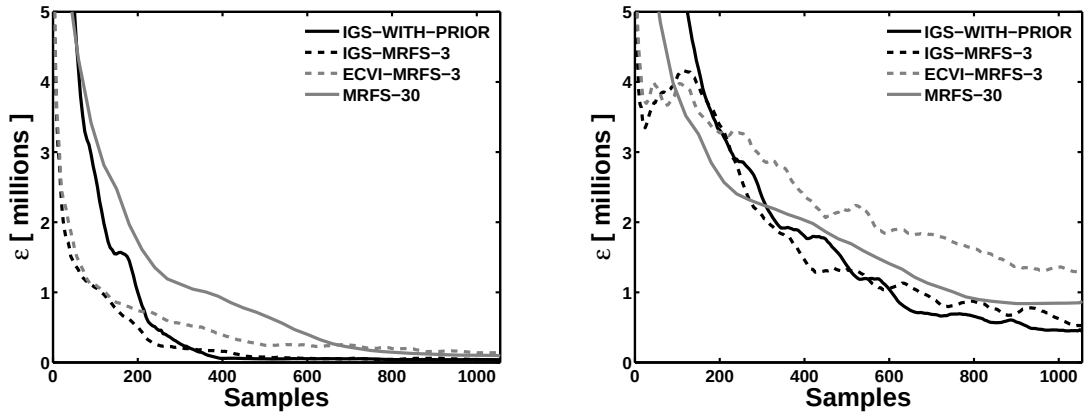

Figure 3: Mean regret (ϵ) in the SCM$\downarrow_3$ 2006 game of the profile returned by the algorithms after a given number of samples when small variance (left) and large variance (right) Gaussian noise is added to the payoffs.

literature. Unlike ECVI, which was an attempt to construct a computationally feasible version of expected value of information, the IGS family of algorithms does not directly resolve to improve the mean estimate of player regret. Instead, the IGS algorithm focuses on improving some distribution over profiles. For example, this distribution could be a distribution of play, a likelihood of PSNE, or the probability that a profile minimizes regret. Optimizing the distribution rather than a point estimate can improve calculations involving the distribution and other heretofore unknown quantities. Consequently, an EVI-based algorithm may not completely capture the decision theoretic-problem underlying the game analysis task.

In addition, IGS does not succumb to a problem that plagues ECVI. That is, ECVI has a tendency to sample *safe* profiles, or precisely, profiles that are not likely to change the current decision in expectation. Thus, ECVI can easily get stuck in local optima and never recover.

Although MRFS was developed and justified under the revealed-payoff model, we have shown that even under noisy payoffs, using MRFS to select initial samples can improve performance early on in the search process. Moreover, we have shown that in some cases MRFS can perform as well as IGS when sampling noisy games.

One significant drawback to MRFS is the constant number of samples per iteration. Given the relative strength MRFS has displayed on the tested classes of games, an interesting future path of study is a dynamic variant of MRFS which takes into account the significance of the deviation comparisons to determine how many times to sample a profile.

Acknowledgments

This work was supported in part by NSF grant IIS-0205435, and the STIET program under NSF IGERT grant 0114368.

9. REFERENCES

[1] R. Arunachalam and N. M. Sadeh. The supply chain trading agent competition. *Electronic Commerce Research and Applications*, 4:63–81, 2005.

[2] J. Eriksson, N. Finne, and S. Janson. Evolution of a supply chain management game for the trading agent competition. *AI Communications*, 19:1–12, 2006.

[3] A. R. Greenwald and J. O. Kephart. Shopbots and pricebots. In *Agent-mediated Electronic Commerce II*, volume 1788 of *Lecture Notes on Artificial Intelligence*. Springer-Verlag, 2000.

[4] P. R. Jordan, C. Kiekintveld, and M. P. Wellman. Empirical game-theoretic analysis of the TAC supply chain game. *Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 1188–1195, May 2007.

[5] P. R. Jordan and M. P. Wellman. Best-first search for approximate equilibria in empirical games. *AAAI-07 Workshop on Trading Agent Design and Analysis (TADA)*, July 2007.

[6] S. Kullback and R. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

[7] D. Monderer and L. S. Shapley. Potential games. *Games and Economic Behavior*, 14:124–143, 1996.

[8] E. Nudelman, J. Wortman, Y. Shoham, and K. Leyton-Brown. Run the GAMUT: A comprehensive approach to evaluating game-theoretic algorithms. In *Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 880–887, New York, 2004.

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

[10] D. M. Reeves, M. P. Wellman, J. K. MacKie-Mason, and A. Osepayshvili. Exploring bidding strategies for market-based scheduling. *Decision Support Systems*, 39:67–85, 2005.

[11] R. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.

[12] S. M. Ross. *Simulation*. Academic Press, 3rd edition, 2001.

[13] S. Russell and E. Wefald. Principles of metareasoning. *Artificial Intelligence*, 49:361–395, 1991.

[14] A. Sureka and P. R. Wurman. Using tabu best-response search to find pure strategy Nash equilibria in normal form games. In *Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1023–1029, Utrecht, 2005.

[15] Y. Vorobeychik, C. Kiekintveld, and M. P. Wellman. Empirical mechanism design: methods, with application to a supply-chain scenario. In *Seventh ACM conference on Electronic Commerce*, pages 306–315, Ann Arbor, 2006.

[16] Y. Vorobeychik and M. P. Wellman. Mechanism design based on beliefs about responsive play (position paper). In *ACM EC-06 Workshop on Alternative Solution Concepts for Mechanism Design*, Ann Arbor, 2006.

[17] W. Walsh, D. Parkes, and R. Das. Choosing samples to compute heuristic-strategy Nash equilibrium. In *Fifth Workshop on Agent-Mediated Electronic Commerce*, 2003.

[18] M. P. Wellman. Methods for empirical game-theoretic analysis (extended abstract). In *Twenty-First National Conference on Artificial Intelligence*, pages 1152–1155, Boston, 2006.

[19] M. P. Wellman, D. M. Reeves, K. M. Lochner, S.-F. Cheng, and R. Suri. Approximate strategic reasoning through hierarchical reduction of large symmetric games. In *Twentieth National Conference on Artificial Intelligence*, pages 502–508, Pittsburgh, 2005.