Probably Almost Stable Strategy Profiles in Simulation-Based Games

Mason Wright and Michael P. Wellman

ABSTRACT

Empirical studies of strategic settings commonly model player interactions under supposed game-theoretic equilibrium behavior, to predict what rational agents might do. But in sufficiently complex settings, analysts cannot solve for exact equilibria, and may resort to solving a restricted game where agents are limited to a tractable subset of strategies. This provides a solution, but one with unclear strategic stability in the original game. We propose a search and evaluation method that can guarantee a well-defined strategic stability property in the profile that it yields, even if only a small subset of possible strategies in a game have been analyzed. The method achieves this result by combining statistical confidence interval estimation, a multiple test correction, and empirical game-theoretic analysis. We also present an extension of the method that more often finds genuine approximate equilibria, by using simulated annealing instead of simple random search for strategy exploration. We demonstrate efficacy in two example settings: the first-price sealed-bid auction, and a cybersecurity game.

KEYWORDS

simulation-based games; empirical game-theoretic analysis

1 INTRODUCTION

In studies of real-world environments with strategic agents, analysts often use game-theoretic equilibrium to predict outcomes of agent interactions. Many games of interest are intractable to solve exactly, due to factors like the large number of strategies and lack of explicit payoff specification. Analysts may cope by considering a *restricted game*, where agents are limited to a relatively small enumerated set of strategies. Payoffs over these strategies can be estimated by simulation, inducing a restricted game model that can be solved for equilibrium. This approach has been applied to areas including models of securities markets [23], social dilemmas [12], space debris removal [10], and credit networks [4].

Such restricted-game studies are informative, but they leave questions about the relevance of solutions found with respect to the original, unrestricted game. In particular, there has been no way to quantify the likelihood that restricted-game findings hold in the original game, even approximately. Indeed, it is generally expected that Nash equilibria found in the restricted games have beneficial strategy deviations in the original game. It would be useful to have some way to characterize stability of restricted-game solutions with respect to the original game, for example in terms of the difficulty of finding beneficial deviations.

We introduce an algorithm that searches for stable profiles in simulation-based games. This algorithm guarantees that when it terminates successfully, there is high statistical confidence the strategy profile returned is *almost stable*, meaning only a small measure of other strategies can be beneficial deviations. In other words, under a specified stochastic search, only a small fraction of trials will yield beneficial deviations. We call this a *probably almost stable* (PAS) guarantee. The degree of statistical confidence and acceptable measure of beneficial deviations can be tuned as desired.

Our procedure uses iterated better-response search to find beneficial deviations. Better-response dynamics, where players iteratively add beneficial deviations to their strategy sets, has been shown to converge to low-regret profiles in various game types [7, 21]. Our procedure also incorporates a statistical confidence interval estimation method for determining whether the current profile meets the stability guarantee or further search is needed. The procedure design is completed by a multiple-test correction for avoiding excessive false positives caused by the sequential-testing nature of the algorithm. Optionally, one can use a black-box optimization procedure like simulated annealing with our procedure to achieve stronger guarantees on the difficulty of finding beneficial deviations, at a cost of increased computation during strategy exploration.

Our contributions are twofold. First, we introduce the new PAS stability concept, which quantifies the likelihood of missing prevalent opportunities for strategic improvement in game-theoretic analysis. Second, we present a novel algorithm for generating strategy profiles in large simulation-based games, that offers a provable guarantee of this form of stability without requiring that most strategies be examined. We give a formal definition of our probably almost stable property and a simple proof of our algorithmic guarantee. We derive the time complexity of our algorithm, in terms of the desired tightness of its statistical guarantee. We evaluate the algorithm's efficacy experimentally on two games: the first-price sealed-bid auction (FPSB), and a cybersecurity game on an attack graph [18]. We show that in both settings, our algorithm yields a high frequency of true positives (i.e., PAS profiles found with high ground-truth success probability), while controlling false positives in accordance with the guarantee. When we incorporate simulated annealing as an improved strategy search method, the frequency of true positives increases dramatically.

2 RELATED WORK

Our approach was inspired in part by work of Bopardikar et al. [2], which defines a search algorithm for *security policies* in two-player, zero-sum simultaneous games. That paper defines a security policy as a mixed strategy and associated payoff for player 1, such that if player 2 plays the best response it finds during a limited random search over deviating strategies, player 1 will obtain at least the given payoff with high probability. We follow this work in reasoning statistically from one player's exploration to derive probabilistic bounds on what the other player can find. However, since we do not assume the game is zero-sum, we cannot (at least

AAMAS-19 Workshop on Games, Agents, and Incentives, accepted, 2019
2019. https://doi.org/doi

in this way) provide guarantees about security value. Rather, we pursue a different probabilistic stability property.

Several works have studied sequential search procedures for Nash equilibrium (NE) in games with many strategies. McMahan et al. [15] introduced the *double-oracle method*, a procedure for iteratively solving two-player, zero-sum games with large strategy sets, using an oracle per agent that best-responds to the equilibrium strategy of the opponent in the current restricted game. Sureka and Wurman [21] combined best-response dynamics with a tabu list to seek pure-strategy NE, for combinatorial auctions. Goldberg [7] found theoretical convergence rates for a better-response dynamics based on randomized local search in a load-balancing game. Schvartzman and Wellman [20] used reinforcement learning to add better-response strategies to a restricted game, finding an NE each time before alternating back to reinforcement learning. Jordan et al. [9] compared strategy exploration procedures in convergence rate to low-regret profiles in the FPSB auction. Recently, Lanctot et al. [11] proposed a generalized iterative method motivated by advances in deep learning from self-play in games.

Our approach relies pivotally on a *strategy exploration function*, which we use to model a player's effort to find a beneficial deviation from a specified strategy profile. All statistical guarantees are relative to this strategy exploration function; a stronger exploration method produces more credible evidence that a proposed strategy profile is an approximate equilibrium. For this reason, we perform experiments with two black-box optimization methods: the naive method of simple random search over a bounded strategy space, and *simulated annealing*, a classical stochastic search method that provably converges to a global optimum with a sufficiently slow annealing schedule. Recent works on black-box optimization have suggested that more complex methods (e.g., Gaussian process bandits) perform better in some problem settings, depending on the number of samples allowed, but with varying results across different problems [8]. We chose simulated annealing as our stronger strategy exploration method, as it is well known, easy to implement, and sufficient to produce noticeably better results than simple random search within our larger method.

3 PROBLEM SETTING

We present our methods for two-player games, with the suggestion that the extension to n players is natural. Let $G = (S, U)$ be a two-player general-sum normal-form game, with strategy sets $S = (S_1, S_2)$ and utility functions $U = (U_1, U_2)$. $U_i(s)$ assigns player i a real-valued payoff for pure-strategy profile $s \in S_1 \times S_2$.

G is a *simulation-based* game, meaning the agents have no direct specification of U, but rather limited access to a *payoff oracle* O. Upon query for a pure-strategy profile s, the oracle responds with $O(s) = (U_1(s), U_2(s))$. The oracle may be said to *simulate* payoffs from the game.

A mixed-strategy profile $\sigma = (\sigma_1, \sigma_2)$ assigns each player a probability distribution over its strategy set. By definition, a mixed-strategy profile σ is a Nash equilibrium of a two-player game, if and only if for all $i \in \{1, 2\}$, $s_i \in S_i$, $E_{s_{-i}\sim\sigma_{-i}} U_i(s_i, s_{-i}) \leq E_{s\sim\sigma} U_i(s)$, where s_{-i} is a pure strategy of the other player, and $\sigma_{-i} \in \Delta_{S_{-i}}$ is the mixed strategy of the other player. More generally, we say that σ is an $\mathcal{E}$-Nash equilibrium, for $\mathcal{E} \geq 0$, if no player can gain more than $\mathcal{E}$ by unilaterally deviating: $E_{s_{-i}\sim\sigma_{-i}} U_i(s_i, s_{-i}) \leq E_{s\sim\sigma} U_i(s) + \mathcal{E}$. We use the term *$\mathcal{E}$-beneficial deviation* to mean any strategy that yields an improved payoff of at least $\mathcal{E}$, as a unilateral deviation from a given reference profile.

We assume an efficient means of finding a mixed-strategy Nash equilibrium (MSNE) in restricted games with sufficiently small strategy sets—perhaps containing a few dozen strategies per agent. We know by Nash's theorem that some mixed-strategy Nash equilibrium must exist in any such game. Solvers implemented in packages such as Gambit can find sample MSNE reliably and efficiently in many practical problems [14].

Our methods are motivated by settings where the strategy sets S_1 and S_2 are too large to explore exhaustively, possibly infinite. We extract a *restricted game* G' from G, written as $G' \subset G$, by restricting players to $S' = (S'_1, S'_2)$, where $S'_1 \subseteq S_1$ and $S'_2 \subseteq S_2$. We must similarly restrict the utility function U' of G' to S', such that U' yields the same result as U where their domains overlap. If restricted game G' has $|S'_1|$ and $|S'_2|$ sufficiently small, by assumption it will be feasible to find some MSNE of G'.

To model strategy exploration, we define a probability distribution D_1 over S_1 and probability distribution D_2 over S_2. Each distribution has full support, that is, for $i \in \{1, 2\}$ and all $s \in S_i$, $D_i(s) > 0$. We assume that agents select strategies to evaluate by sampling from these distributions in an i.i.d. manner. For example, in this study, we perform experiments where D_i is defined implicitly by either simple random search or simulated annealing, over continuous strategy spaces.

4 PROBABLY ALMOST STABLE PROFILE SEARCH

Suppose player 1 has found an $\mathcal{E}$-MSNE, $\sigma = (\sigma_1, \sigma_2)$, in a restricted game $G' \subset G$. It is possible that player 2 might, through a limited random search, find some strategy s'_2 that player 1 has not considered, from $G \setminus G'$, such that $U_2(\sigma_1, s'_2) \geq U_2(\sigma) + \mathcal{E}$—that is, an $\mathcal{E}$-beneficial deviation for player 2. We would like to be able to say this is unlikely: that, if player 2 samples a pure strategy from distribution D_2, the probability of obtaining an $\mathcal{E}$-beneficial deviation is no greater than ϵ, with $0 < \epsilon \ll 1$.

Definition 4.1. A strategy profile σ is *ϵ-almost stable* for player 1 with respect to D_2 and $\mathcal{E}$, if

$$\Pr\big(U_2(\sigma_1, s'_2) > U_2(\sigma) + \mathcal{E}\big) \leq \epsilon, \text{ for } s'_2 \sim D_2.$$

The almost-stability concept captures a bound on the measure of beneficial deviations in the strategy space. Because our evidence about the overall space of strategies comes exclusively from sampling, the best that we can do is to show that profile σ is almost stable with high probability, say at least $(1 - \delta)$, with $0 < \delta \ll 1$. If a profile σ is accepted as almost stable by a statistical test that has a false positive rate of at most δ for any input, we say that σ is *probably almost stable*. Let $p \equiv \Pr(U_2(\sigma_1, s'_2) > U_2(\sigma) + \mathcal{E})$.

Definition 4.2. A strategy profile σ is *probably ϵ-almost stable* for player 1 with respect to D_2, $\mathcal{E}$, and δ if it is accepted by a statistical hypothesis test T, such that whenever $p > \epsilon$,

$$\Pr\big(T(\sigma) = \textit{accept}\big) \leq \delta,$$

with respect to the randomness in the statistical test.

We might like to model the opponent as drawing M i.i.d. samples from D_2, seeking a probably almost stable guarantee reflecting the likelihood of *any* sample being an $\mathcal{E}$-beneficial deviation. This can be achieved by adjusting the deviation probability, as $\epsilon \leftarrow 1-(1-\epsilon)^{1/M}$. However, such adjustment may dramatically increase the number of samples required to establish the guarantee.

4.1 Using the Clopper-Pearson confidence interval

The Clopper-Pearson confidence interval is a classical statistical tool; the summary below (for the upper bound only) is based on the original paper [5]. We use this method to derive a bound on the probability of any sampled deviation being $\mathcal{E}$-beneficial, based on the observed frequency in a sample. The bound is conservative, even when no successes are observed [1, 3].

Given a binomial random variable of unknown probability p, we observe X successes in N trials. The Clopper-Pearson procedure, given a desired confidence level $\delta \in (0, 1]$ and sample count N, provides an upper bound $\bar{p}^{\delta}_{N}(X)$ on p, such that for any $p \in (0, 1]$, the risk of error (i.e., $\Pr(\bar{p}^{\delta}_{N}(X) < p)$) is at most δ.

For the desired δ and sample count N, and for any p, let $\underline{x}^{\delta}_{N}(p)$ be the greatest integer in $\{-1, \ldots, N-1\}$ such that $\Pr(X \leq \underline{x}^{\delta}_{N}(p) \mid p) \leq \delta$. For any $X \in \{0, \ldots, N-1\}$, let $\bar{p}^{\delta}_{N}(X)$ be the least value $p' \in [0, 1]$ such that $\underline{x}^{\delta}_{N}(p') = X$. Define $\bar{p}^{\delta}_{N}(N) = 1$. The Clopper-Pearson upper bound on p is simply $\bar{p}^{\delta}_{N}(X)$.

It has been shown [5] that given $\delta \in (0, 1]$ and $N \geq 1$, for all $p \in [0, 1]$,

$$\Pr\left(\bar{p}^{\delta}_{N}(X) < p\right) \leq \delta. \tag{1}$$

In our procedure, we select a sample count N as small as possible such that if no successes are observed in N trials, the Clopper-Pearson upper bound, $\bar{p}^{\delta}_{N}(0)$, will be at most ϵ. As Thulin [22] notes, in the case where $X = 0$ trials are successful, we can compute the upper bound of the Clopper-Pearson interval as $\bar{p}^{\delta}_{N}(0) = 1 - \delta^{1/N}$. Rewriting to solve for the required number of trials to guarantee $\bar{p}^{\delta}_{N}(0) \leq \epsilon$, we find:

$$N = \left\lceil \frac{\log(\delta)}{\log(1-\epsilon)} \right\rceil. \tag{2}$$

In case a lower false negative rate is desired (i.e., fewer almost-stable profiles rejected), our procedure could be modified to select N such that one or more successes are allowed in N trials, with the general approach being otherwise unchanged.

4.2 Hypothesis test for probably almost stable profiles

Here we present a straightforward hypothesis test for evaluating whether a mixed-strategy profile $\sigma = (\sigma_1, \sigma_2)$ is probably almost stable for player 1 with given parameters. (To guarantee that neither player is likely to find an $\mathcal{E}$-beneficial deviation by sampling, we could simply run Algorithm 1 with each player in turn as focal agent.) If $p > \epsilon$, (i.e., random draws are too likely to be $\mathcal{E}$-beneficial deviations), this test will reject profile σ with probability at least $(1 - \delta)$.

Algorithm 1 PAS-Single

Require: $\epsilon \in (0, 1), \delta \in (0, 1), \sigma \in \Delta_{S_1} \times \Delta_{S_2'}, D_2 \in \Delta_{S_2}, U_2 : \Delta_{S_1} \times \Delta_{S_2} \to \mathbb{R}, \mathcal{E} \geq 0$

$N \leftarrow \arg\min_{\ell \in \{1, \ldots\}} : \bar{p}^{\delta}_{\ell}(0) \leq \epsilon$
for N trials **do**
  Sample a player-2 pure strategy $s_2 \sim D_2$
  if $U_2(\sigma_1, s_2) > U_2(\sigma) + \mathcal{E}$ **then**
    return $(\textit{reject}, s_2)$
return $(\textit{accept}, \perp)$

In presenting Algorithm 1 we use the notation $\bar{p}^{\delta}_{\ell}(X)$ to mean the upper bound on p where ℓ is the binomial sample count, and δ the desired confidence level. We reject the hypothesis that profile σ is probably almost stable if a single $\mathcal{E}$-beneficial deviation is found in N trials, and otherwise we accept. We also return the pure strategy s_2 that $\mathcal{E}$-beneficially deviated, or $\perp$ if none was found. By noting that $\sigma \in \Delta_{S_1} \times \Delta_{S_2'}$, we emphasize that in the input profile player 2 plays only a small (finite) subset of its full strategy set with positive probability.

Proposition 4.3. *For input $(\epsilon, \delta, \sigma, D_2, U_2, \mathcal{E})$, if Algorithm 1 returns* accept, *then σ is probably ϵ-almost stable at confidence level δ for distribution D_2 and $\mathcal{E}$.*

Proof. Let T represent Algorithm 1. We want to show that $\Pr(T(\sigma) = \textit{accept} \mid p > \epsilon) \leq \delta$. Consider any σ such that $p > \epsilon$. $T(\sigma) = \textit{accept}$ only if the success count $X = 0$ in N trials. By construction, $\bar{p}^{\delta}_{N}(0) \leq \epsilon$. Thus, if $T(\sigma) = \textit{accept}$ and $p > \epsilon$, $\bar{p}^{\delta}_{N}(X) < p$. By the Clopper-Pearson bound (1), $\Pr(\bar{p}^{\delta}_{N}(X) < p) \leq \delta$. □

Proposition 4.3 holds because the Clopper-Pearson estimator limits the false positive rate of Algorithm 1 to δ, even if $p \approx \epsilon$. One drawback of the estimator is that it produces a high false negative rate when $p \approx \epsilon$. Specifically, the probability of a false negative in Algorithm 1, when $p \leq \epsilon$, is $1 - (1 - p)^N$. This is simply the probability that at least one sampled deviation will be $\mathcal{E}$-beneficial. In the worst case where $p = \epsilon$, the false negative rate equals $(1 - \delta)$. Since many profiles σ that should be accepted by Algorithm 1 have $p \ll \epsilon$, the false negative rate is often much lower than $(1 - \delta)$.

5 COMBINED BETTER-RESPONSE SEARCH AND SEQUENTIAL HYPOTHESIS TEST

Algorithm 1 can evaluate whether a given profile σ is probably almost stable. This can be employed within a broader method to *search* for such a profile. Suppose Algorithm 1 returns $(\textit{reject}, s_2)$ for profile σ. We can append the pure strategy s_2 that beneficially deviates from σ to the old restricted strategy set S', to form enlarged strategy set S''. Next, we query our payoff oracle O for the payoffs of every new pure strategy profile in the enlarged restricted game G''. We can then use our game solver to find a MSNE σ'' of G''. Finally, we test profile σ'' to determine whether it is probably almost stable in original game G.

If the new profile σ'' is found to be probably almost stable in original game G, we stop with a successful result. Otherwise, we add the beneficially deviating pure strategy to the restricted strategy set

and repeat, until either we succeed and stop, or some predetermined number of iterations is reached and we fail.

Note that this procedure may invoke the statistical hypothesis test many times. Thus in order to guarantee probably almost stability at a given confidence level, we need to apply a multiple-comparison adjustment to the level δ employed at each iteration of Algorithm 1.

Algorithm 2 combines a better-response dynamics search for stable profiles with a sequential hypothesis testing procedure that accounts for multiple comparisons. We aim to limit the familywise error rate (FWER) over all hypothesis tests in the sequence to a given confidence level δ. This means that for any game G, initial restricted strategy set S_2', deviation probability ϵ, confidence level δ, equilibrium candidate σ, distribution D_2, and deviation tolerance $\mathcal{E}$, the probability of returning $(accept, \sigma')$, when the true probability of beneficial deviation from σ' under D_2, $\mathcal{E}$, is greater than ϵ, is at most δ.

Algorithm 2 PAS-Sequential

Require: $\epsilon \in (0,1), \delta \in (0,1), \sigma \in \Delta_{S_1} \times \Delta_{S_2'}, D_2 \in \Delta_{S_2}, U : \Delta_{S_1} \times \Delta_{S_2} \to \mathbb{R}^2, \mathcal{E} \geq 0, K \in \{1, \dots\}, \vec{\alpha} \in \mathbb{R}_+^K : \sum \alpha_k = \delta, S_2' \subset S_2$

$\sigma' \leftarrow \sigma$, the current restricted-game MSNE
for iteration $k \in \{1, \dots, K\}$ **do**
  $N_k \leftarrow \arg\min_{\ell \in \{1, \dots\}} : \bar{p}_\ell^{\alpha_k}(0) \leq \epsilon$
  $s \leftarrow \perp$, the $\mathcal{E}$-beneficial deviation
  for N_k trials **do**
    Sample a player-2 pure strategy $s_2 \sim D_2$
    if $U_2(\sigma_1', s_2) > U_2(\sigma') + \mathcal{E}$ **then**
      $s \leftarrow s_2$
      break
  if $s = \perp$ **then**
    return $(accept, \sigma')$
  else
    $S_2' \leftarrow S_2' \cup \{s\}$
    $\sigma' \leftarrow$ *solve*(S_1, S_2', U), an MSNE for the new player-2 strategy set
return $(reject, s)$

Note that function *solve*(S_1, S_2', U) in Algorithm 2 returns any MSNE of the restricted game on strategy set (S_1, S_2').

Algorithm 2 uses a Bonferroni correction for multiple comparisons testing, to cap the familywise error rate at δ. K is the maximum number of strategy exploration rounds to perform. $\vec{\alpha} \in \mathbb{R}_+^K$ is an α-spending vector, indicating the confidence level to be used in each iteration, where the confidence levels sum to δ. By the union bound, we know that if the probability of a false positive in each round k is at most α_k, the probability of any false positive is at most $\sum \alpha_k = \delta$. Therefore, because Algorithm 1 guarantees probably almost stable profiles when it returns *accept*, it follows that Algorithm 2 does as well. Algorithm 2 can make this further guarantee, compared to the single-round Algorithm 1, because in any round k it requires a larger number of tests N_k to guarantee $\bar{p}_{N_k}^{\alpha_k}(0) \leq \epsilon$.

Proposition 5.1. *For any input $(\epsilon, \delta, \sigma, D_2, U, \mathcal{E}, K, \vec{\alpha}, S_2')$, if Algorithm 2 returns $(accept, \sigma')$, σ' is probably ϵ-almost stable at confidence level δ for distribution D_2 and $\mathcal{E}$.*

Proof. For each of up to K rounds of Algorithm 2, let $R^k(\sigma^k) \in \{accept, reject\}$ be the result of round k, on current profile σ^k. For any profile σ'', let $\Pr(R^k(\sigma'') = accept)$ be the acceptance probability of the test in Algorithm 2 for round k, given α_k. For any profile σ'', let $p(\sigma'')$ equal the $\mathcal{E}$-beneficial deviation probability under distribution D_2. Let us define hypothesis test $T(\sigma'')$ as the test that, for any profile σ'', returns *accept* if any round k of Algorithm 2 returns $(accept, \sigma'')$, and *reject* otherwise.

We want to show that for any profile σ'', $\Pr(T(\sigma'') = accept \mid p(\sigma'') > \epsilon) \leq \delta$.

By the union bound,

$$\Pr\big(T(\sigma'') = accept\big) \leq \sum_{k=1}^{K} \Pr\big(R^k(\sigma'') = accept\big).$$

By construction, in each round k of Algorithm 2, for any profile σ'', if $p(\sigma'') > \epsilon$, then $\Pr(R^k(\sigma'') = accept) \leq \alpha_k$. This is because each round of Algorithm 2 essentially implements Algorithm 1 with measure tolerance ϵ and confidence level α_k. Thus, by the union bound, if $p(\sigma'') > \epsilon$, $\Pr(T(\sigma'') = accept) \leq \sum_{k=1}^{K} \alpha_k = \delta$. □

Corollary 5.2. *The probability that Algorithm 2 returns any tuple $(accept, \sigma')$ such that $p(\sigma') > \epsilon$ is at most δ.*

5.1 Asymptotic cost of Algorithm 1

The order of growth of time and space required by Algorithm 1 is modest. The algorithm takes inputs ϵ, the allowed deviation probability, and δ, the confidence level; the algorithm's runtime can be represented by N, the worst-case sample count. The space complexity of Algorithm 1 is constant, because the algorithm merely needs to track the expected payoff of the best deviation found so far and how many deviations have been sampled.

The time complexity can be conveniently expressed as a function of $\frac{1}{\epsilon}$, the expected samples per $\mathcal{E}$-beneficial deviation, and $\frac{1}{\delta}$, the expected trials before a false positive. Algorithm 1 has runtime N that is $O(\frac{1}{\epsilon})$ and $O(\log \frac{1}{\delta})$. The bound for $\frac{1}{\delta}$ is trivial, based on (2). We can derive the bound for $\frac{1}{\epsilon}$ by letting $y = \frac{1}{\epsilon}$ and noting that an upper bound on N is proportional to $f(y) = -(\log \frac{y-1}{y})^{-1}$; we then take the derivative of f with respect to y, and show that its limit is 1 as y approaches infinity. This shows that N grows with y proportionally to a linear function in the limit. Thus, the order of growth in the strategy count N that the analyst must sample is proportionate to the desired bound on the number of samples an agent is expected to require to find an $\mathcal{E}$-beneficial deviation, $\frac{1}{\epsilon}$.

For a numerical example, with $\epsilon = 0.01$ and $\delta = 0.1$, $N = 230$ samples are needed in the worst case. If instead $\epsilon = 10^{-5}$, $N =$ 230,258 samples would be needed. Note that $\frac{N}{\epsilon}$ is roughly constant for a given δ, so the analyst's computational needs scale evenly with the agent's power to explore.

5.2 Asymptotic cost of Algorithm 2

The worst-case time required to run Algorithm 2 is $N \times K$, where N is the maximum sample count per round, and K is the maximum round count. The time complexity of Algorithm 2, like that of Algorithm 1, is $O(\frac{1}{\epsilon})$ and $O(\log \frac{1}{\delta})$. Moreover, the time complexity $N \times K$ is $O(K \log K)$ with respect to the round count. To see this, observe that there is an $O(K)$ factor due to the K term in $N \times K$ and

an $O(\log K)$ factor due to the division of δ by K in (2) for finding N, when we divide the δ budget into K equal parts for the various rounds.

For instance, if $\epsilon = 0.01$, $\delta = 0.1$, and $K = 3$, Algorithm 2 requires $N \times K = 1{,}017$ samples in the worst case. Even if K were set as high as 1,000, the worst-case sample count would be 917,000; with this many strategies, the bottleneck would likely be the Nash equilibrium solver, not in the search process itself. The sample complexity $N \times K$ grows proportionally to $\frac{1}{\epsilon}$.

6 EXPERIMENTS

We perform experiments on two classes of two-player, general-sum, normal form games from the game theory literature. We study the first-price sealed-bid auction (FPSB) and a cybersecurity game played on an attack graph, from recent literature [17]. In the FPSB game, we consider a parameterized strategy, where players bid a constant fraction of their value for the item being sold. This strategy family includes equilibria of the full game, and has been adopted for analytical convenience in prior work on FPSB auctions [6, 19].

Each experiment begins with a restricted strategy set. We execute single-pass Algorithm 1 or sequential Algorithm 2, and analyze the results based on frequency of true and false positives, as well as acceptance rate conditional on true beneficial deviation probability p (as estimated via sampling). We also repeat the sequential experiments with both strategy exploration methods under consideration: simple random search and simulated annealing.

We aim to test whether the Algorithms 1 and 2 empirically satisfy their theoretical guarantees, even for values of $p \approx \epsilon$. In addition, we aim to evaluate whether the frequency of true positives is reasonably high, such that the algorithms are likely to be useful for finding stable profiles, instead of rejecting almost all profiles due to excessive conservatism. Our results also demonstrate the performance of our algorithms when a relatively sophisticated search process is used to seek beneficial deviating strategies.

6.1 First-price sealed-bid auction game

In the two-player FPSB auction game, one item is auctioned to the higher bidder of player 1 and player 2, and the winner pays its bid. Our version of the game allows each player i to choose a bid fraction $c_i \in [0, 1]$ (before learning the player's value for the item). Each player is then assigned a private value v_i for the item, drawn i.i.d. from $U(0, 1)$, and bids $c_i v_i$. The higher bidder wins the item, earning utility $v_i(1 - c_i)$; the other player earns zero utility.

The unique Nash equilibrium in 2-player FPSB is for player i to play the pure strategy $c_i = \frac{1}{2}$ [6, 9]. In a restricted version of the FPSB auction game, allowing only a finite set of bid factors $S' = \{c_1, \ldots, c_q\}$, the equilibrium solution is less simple. In fact, $c = \frac{1}{2}$ is not necessarily a beneficial deviation from a Nash equilibrium of such a restricted game. This leads to interesting complications in the iterated better-response dynamics of such restricted games.

Some known properties of the FPSB auction game facilitate our analysis. Suppose that player 2 plays pure strategy c_2, and we want to evaluate the expected payoff for player 1 of playing pure strategy c_1. This utility is given by:

$$\mathbb{E}_{v_1, v_2 \sim U(0,1)}(U_1(c_1, c_2)) = \begin{cases} 0 & \text{if } c_1 = 0 \\ \frac{1-c_1}{2} & \text{else if } c_2 = 0 \\ \frac{(1-c_1)c_1}{3c_2} & \text{else if } c_1 \leq c_2 \\ (1 - c_1)\frac{3c_1^2 - c_2^2}{6c_1^2} & \text{otherwise.} \end{cases} \quad (3)$$

Using (3), we can determine whether c_1 is a beneficial deviation from any finite mixed strategy for player 2. Moreover, we can immediately give a minimal example of a restricted game where $c = \frac{1}{2}$ is not a beneficial deviation. Consider the restricted game where the only legal strategy is $c_a = \frac{1}{3}$; in the unique Nash equilibrium, the expected payoff is $\frac{2}{9} = \frac{24}{108}$. A player deviating to play $c_b = \frac{1}{2}$ would receive expected payoff $\frac{23}{108}$, which is lower.

For our experiments on the FPSB auction, we begin each restricted game with a shared set S' of 10 pure strategies c available for the players, selected i.i.d. from $U(0, 1)$. We use (3) to build the payoff matrix for all pairs of pure strategies in S'. Then we use Gambit's implementation of the extreme points method [13] to find MSNEs of the resulting game, and select uniformly among them to set σ' for the next iteration of Algorithm 2.

6.2 Cybersecurity game

We also examine an attack-graph cybersecurity game from a recent study [17]. In brief, the game represents an adversarial but not zero-sum interaction between an attacker and a defender, modeled using Bayesian attack graphs [16]. The defender agent in our experiments varies along three parameters, adjusting how many nodes are typically defended, and how randomly or greedily to act. The game is simulation-based, meaning that profile payoffs are estimated by sampling the results of a simulator. This game serves as a realistic application domain for the PAS algorithms, as it is a simulation-based game where it is costly to obtain payoff samples, and there is no known, efficient method of finding exact Nash equilibria in the unrestricted strategy space.

6.3 Simulated annealing

In some experiments, we use the classical method of simulated annealing as an improved alternative to simple random search for strategy exploration. Here we briefly describe how our implementation of simulated annealing is configured.

We model the strategy space for the deviating agent as $S_2' \in [0, 1]^d$, where $d = 1$ for FPSB and $d = 3$ for the cybersecurity game. (Strategy vectors in this space can be mapped to and from the original strategy space S_2.) We decide on a round count κ of strategies to examine in each run of simulated annealing, using $\kappa = 50$ for FPSB, and due to computational limitations, $\kappa = 5$ in the cybersecurity game.

Each run of simulated annealing begins with an i.i.d. random sample over S_2', in our experiments a uniform random sample; it also has an i.i.d. random seed. From there, we employ a truncated Gaussian as our distribution for sampling a neighbor of the current strategy vector. That is, for each dimension in d, we use an independent Gaussian draw centered on the current strategy value, with some given variance, using rejection sampling to ensure the result is in $[0, 1]$. We use a variance of 0.003 for the FPSB game,

0.03 for the cybersecurity game. (We round all randomly generated strategy vectors to 6 decimal places.)

For the temperature schedule, we anneal the temperature τ linearly over the number of samples analyzed so far, from a maximum of 1.0 in FPSB or 15.0 in the cybersecurity game, to zero. At each step, we update the current strategy vector to the new one being analyzed, if: (a) the new strategy has higher expected payoff, or (b) with probability $\exp((u' - u)/\tau)$, where u' is the new strategy's expected payoff, and u is the current strategy's expected payoff. We return the parameters that yield the highest expected payoff, not necessarily the final parameters settled on by the search process.

7 RESULTS

7.1 FPSB auction

We consider the FPSB auction with 10 randomly generated initial strategies, maximum deviation probability allowed $\epsilon = 0.05$, and FWER $\delta = 0.1$. We begin by considering the special case where $\mathcal{E} = 0$, so any payoff improvement is sufficient for an $\mathcal{E}$-beneficial deviation. Over these FPSB games, the mean probability of a sampled strategy being a beneficial deviation from the initial MSNE was 10.4%, with a median of 7.4%. Therefore, in the majority of these FPSB games, the initial restricted game's MSNE is not almost stable, where $\epsilon = 0.05$; more strategies would have to be added to the restricted game to produce an almost-stable profile.

We used a high value of $\epsilon = 0.05$ in the experiments that empirically validate our algorithm's accuracy, because a higher ϵ is more economical, and the value of ϵ should not affect our algorithm's correctness. But we reiterate that the runtime of Algorithm 2 grows only linearly in $\frac{1}{\epsilon}$, such that even if we had used $\epsilon = 10^{-4}$, for example, our experiments would remain computationally feasible. Moreover, we used a low value of $K = 3$ because this is appeared to be the lowest K that would clearly show the need for multiple-tests correction, and a higher K would not yield dramatically more convincing results. The sample count $N \times K$ required by Algorithm 2 grows as $O(K \log K)$, so a larger iteration limit like $K = 10$ would be easily manageable. (Indeed, we also conducted a follow-up experiment with $\epsilon = 0.0005$ and $K = 12$.)

As shown in Figure 1, the empirical accept rate of Algorithm 1 is less than δ for all true beneficial deviation probabilities $p > \epsilon$. This is the guarantee we ensure when we certify profiles as probably almost stable. The plot is based on 400 randomly-generated FPSB games, for which the PAS-single algorithm was run 100 times per game. Overall, the PAS-single algorithm accepted a profile as stable in 13.7% of cases, with frequencies as follows: true positive 12.6%, true negative 64.9%, false positive 1.1%, false negative 21.4%. All of our results are summarized in Table 1.

Next, we demonstrate the necessity of controlling for multiple comparisons. We show what happens if we run the sequential Algorithm 2 with a maximum of $K = 3$ iterations, on the same FPSB game, but without adjusting the α used per iteration by a factor of $\frac{1}{K}$ relative to the value what would be used in single-pass Algorithm 1. Specifically, we let $\alpha_i = \delta$ for all rounds, essentially repeating the single-pass Algorithm 1 up to 3 times. Figure 2 (left) shows that when the true beneficial deviation probability p is slightly greater than ϵ, this sequential procedure will incorrectly accept the profile as almost stable, more than δ fraction of the time. Therefore, the procedure fails to guarantee probably almost stable results when it accepts. This procedure achieves a false positive rate of 3.5%, which is below δ. But for $p \approx \epsilon$, the procedure fails to guarantee an acceptably low false positive rate.

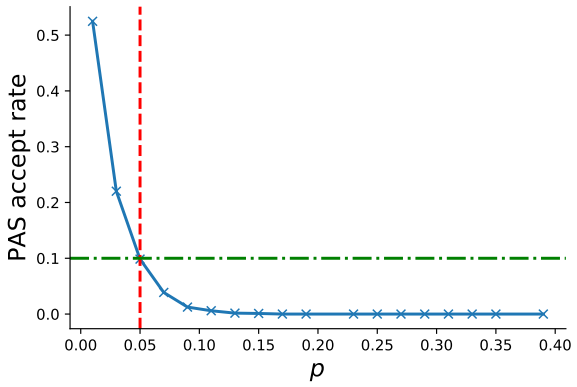

Figure 1: Accept rate in FPSB of Algorithm 1 vs. probability p of beneficial deviation. Horizontal line is $\delta = 0.1$, vertical line $\epsilon = 0.05$. Accept rate is below δ for all $p > \epsilon$.

Finally, we show Algorithm 2 with $\sum \alpha_i = \delta$ corrects the multiple comparisons problem. Again, we run sequential Algorithm 2 for up to $K = 3$ iterations, but with $\alpha_i = \frac{\delta}{3}$ in each round. Figure 2 (right) shows when the true beneficial deviation probability p is greater than ϵ, the accept probability is less than δ, meaning the probably almost stable guarantee is satisfied when Algorithm 2 accepts. The PAS-sequential procedure accepts a profile as almost stable in 40.0% of trials, much better than the 13.7% that was achieved by PAS-single with the same ϵ and δ values. Overall, PAS-sequential produces these frequencies: true positive 39.1%, true negative 24.0%, false positive 0.8%, false negative 36.1%. (These rates are computed over only the final iteration's profile from each trial, not including the profiles from earlier iterations within a trial.) On average, PAS-sequential terminated after 2.3 of a possible 3 iterations of strategy search, using a median of 3 iterations.

Notice PAS-sequential is more successful at finding a stable profile than PAS-single: PAS-single yields a frequency of positives of only 13.7%, while the 3-round version of PAS-sequential yields 39.1%. This may be because, as previously mentioned, better-response dynamics tends to produce increasingly stable profiles.

Figure 3 shows that the probability of finding a beneficial deviation from the current restricted-game MSNE σ empirically decreases in each round of Algorithm 2. In a typical run, the initial strategy set does not yield an almost-stable MSNE at $\epsilon = 0.05$. With each round of the algorithm, however, the distribution of beneficial deviation probabilities shifts to the left, as desired.

To show that Algorithm 2 is feasible with much lower ϵ and higher K, we ran a follow-up experiment with $\epsilon = 0.0005$, $K = 12$, and other settings as before. Over 400 trials, we achieved the following frequencies: true positive 38.5%, true negative 18.7%, false positive 0.0%, false negative 42.8%. The mean rounds before termination was 10.91, of a maximum 12 possible.

Figure 4 shows the acceptance rate of Algorithm 2 as a function of the true beneficial deviation probability p, when $\epsilon = 0.0005$, $\delta = 0.1$,

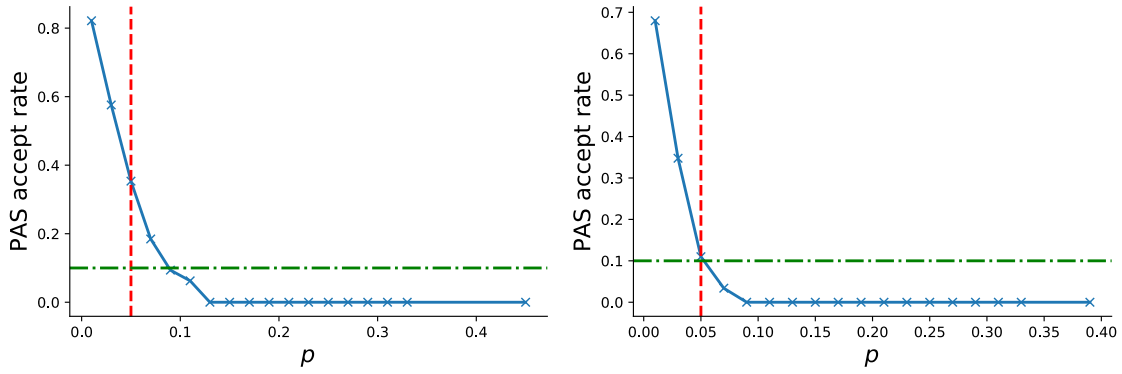

Figure 2: Accept rate in FPSB of Algorithm 2 vs. probability p of finding a beneficial deviation, with up to $K = 3$ rounds. Left: without multiple comparisons control ($\alpha_i = \delta$). Right: with control ($\alpha_i = \frac{\delta}{3}$). Horizontal line is $\delta = 0.1$; vertical line $\epsilon = 0.05$.

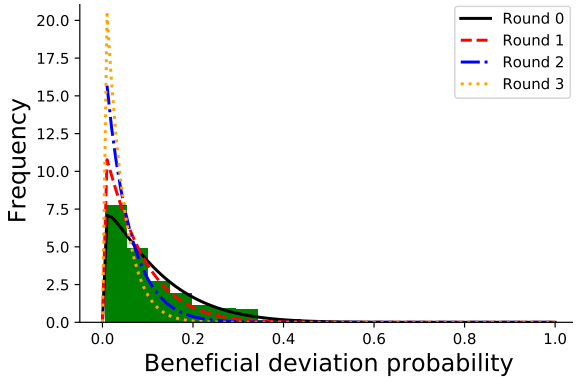

Figure 3: Empirical distribution in FPSB of deviation probability; random strategies $\sim D_2 = U(0, 1)$, vs. MSNE after K rounds of Algorithm 2. Histogram shows Round 0.

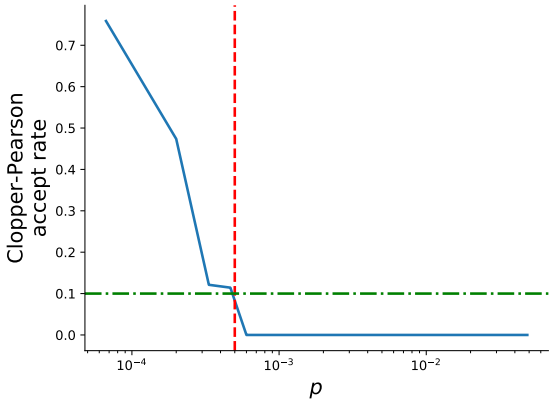

Figure 4: Accept rate in FPSB of Algorithm 2 vs. probability p of finding a beneficial deviation, where $\alpha_i = \frac{\delta}{12}$ for up to $K = 12$ rounds. Horizontal line is $\delta = 0.1$; vertical line $\epsilon = 0.0005$. X-axis has log scale.

$\mathcal{E} = 0.0$, and $K = 12$. Observe that, as desired, for $p > 0.0005$, the acceptance rate is below 0.1. The figure is based on 400 trials.

This experiment took considerably longer to complete than the others, due to the larger number of candidate deviations to evaluate per round, and due to the more challenging equilibrium-finding problems, with their many similar pure strategies.

7.1.1 FPSB with simulated annealing. Here we present results for the FPSB auction, when simulated annealing over $\kappa = 50$ steps is used instead of simple random search as the strategy exploration method. In this experiment, we use a similar configuration to the above, with 10 strategies initially available, $\epsilon = 0.05$, and $\delta = 0.1$.

Because simulated annealing is much better at finding beneficial deviations than simple random search, we increased the maximum strategies added, K, to 4. We also used a nonzero $\mathcal{E}$ of 0.0001.

Over 400 sampled games with different, randomly-generated initial strategy sets, our procedure returned an *accept* result (i.e., a supposedly stable profile) in 382 cases, or with frequency 0.955. To evaluate the ground-truth likelihood that simulated annealing would find an $\mathcal{E}$-beneficial deviation from each returned profile, we ran 200 independent trials of simulated annealing. In every trial, the follow-up test frequency of finding $\mathcal{E}$-beneficial deviations was below δ for positive results and above δ for negative results, yielding a frequencies of true positives of 95.5% and frequency of true negatives of 4.5%. These results demonstrate that in a simple domain like the two-player FPSB auction, our iterated profile search method combined with simulated annealing can perform remarkably well. (Note that the reasons for the nearly-ideal performance include the positive $\mathcal{E}$ tolerance, interacting with the smooth payoff function of the FPSB game, and the rounding of sampled strategies to 6 decimal places.)

7.2 Cybersecurity game

In the cybersecurity game with simple random strategy search, we begin with a fixed set of 10 strategies for the attacker and 12 for the defender. The attacker plays a mixture over this strategy set, while the defender explores randomly sampled alternative strategies. The defender's distribution D_2 may be viewed as sampling

	ϵ	K	TP	TN	FP	FN
FPSB/Alg. 1/rand.	0.05	-	12.6%	64.9%	1.1%	21.4%
FPSB/Alg. 2/rand.	0.05	3	39.1%	24.0%	0.8%	36.1%
FPSB/Alg. 2/rand.	5e-4	12	38.5%	18.7%	0.0%	42.8%
FPSB/Alg. 2/s. a.	0.05	4	95.5%	4.5%	0.0%	0.0%
Cyber./Alg. 2/rand.	0.05	3	26.0%	23.4%	1.0%	49.6%
Cyber./Alg. 2/s. a.	0.05	10	93.0%	1.0%	0.0%	6.0%

Table 1: Resulting frequencies of True Positive, True Negative, False Positive, and False Negative.

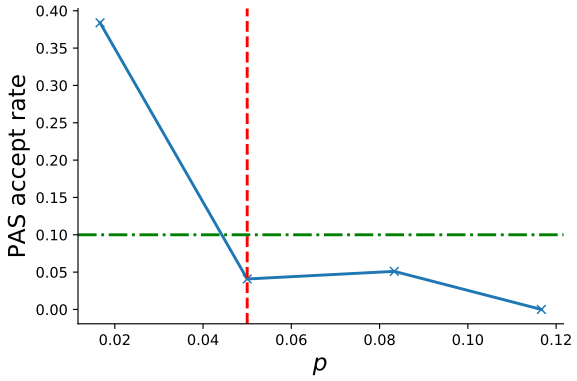

Figure 5: Accept rate in cybersecurity game of Alg. 2 vs. true probability p of finding a beneficial deviation, where $\alpha_i = \frac{\delta}{3}$ for up to $K = 3$ rounds. Horizontal line is $\delta = 0.1$; vertical line $\epsilon = 0.05$. As desired, accept rate is below δ for $p > \epsilon$.

parameterizations $(a, b, c) \in \mathbb{R}^3$ for a heuristic strategy that takes 3 parameters, where each parameter is drawn i.i.d. from $U(0, 1)$. We verified empirically that under the initial strategy set, the beneficial deviation probability is greater than $\epsilon = 0.05$, so the initial restricted game's MSNE is *not* almost stable. As before, we use $\delta = 0.1$. We begin again with the special case where $\mathcal{E} = 0$, so all payoff improvements count as $\mathcal{E}$-beneficial deviations.

Figure 5 shows the accept rate of Algorithm 2 versus an estimate of the true beneficial deviation probability in the cybersecurity game. Here we estimate the true beneficial deviation probability for the final profile σ when Algorithm 2 terminates by sampling 400 deviations from D_2 and finding the sample mean payoff of each deviation over 250 simulations. We collected data from 700 independent runs of Algorithm 2, requiring about 400 hours of computation time, the majority of which was used to estimate the true beneficial deviation probability of each terminal profile σ.

Note that as shown in Figure 5, for all $p > \epsilon$, Algorithm 2 produces an acceptance rate less than δ, as desired. The figure shows a less smooth curve than for FPSB, with an acceptance rate with $p \approx \epsilon$ markedly below δ, perhaps because our estimation procedure for p did not obtain enough samples to generate accurate estimates of the true beneficial deviation probability. It may be that the Bonferroni correction is too conservative in this setting, which is likely if the hypothesis tests' ground-truth results are positively correlated. In the cybersecurity game, Algorithm 2 yields frequencies of true positive 26.0%, true negative 23.4%, false positive 1.0%, false negative 49.6%. Algorithm 2 terminated after a mean 2.89, median 3 rounds, of up to 3 possible.

7.2.1 Cybersecurity game with simulated annealing. In the cybersecurity game, we test simulated annealing of $\kappa = 5$ steps as the strategy exploration method. The experiments begin with a set of eight attacker strategies and 50 defender strategies. As before, we set $\epsilon = 0.05$ and $\delta = 0.1$. We use maximum round count $K = 10$. Due to computational constraints, we reduce the sample count for each payoff estimate to 100, and sample only 100 strategy deviations via simulated annealing to estimate the ground-truth probability of beneficial deviation. To reduce the occurrence of spurious deviations being found, due to the lower sample count used for payoff estimation, we increase the $\mathcal{E}$ threshold to 1.3 in this experiment. Each run of the experiment consumed between 40 and 120 hours on one Intel Xeon CPU, depending on the number of rounds required.

We performed 97 runs of Algorithm 2 in the cybersecurity setting with simulated annealing. The mean number of stages before convergence was 4.4 out of the maximum 10. The fraction of runs that returned *accept* was 0.93. All accepting runs appear to be true positives, with a mean estimated ground-truth success probability of 0.009. However, 0.06 fraction of all results were false negatives, and 0.01 were true negatives. (One *reject* run failed to terminate properly and is excluded from results.) More encouragingly, the mean estimated ground-truth success probability of runs that returned *reject* was 0.037, much higher than for the runs that returned *accept*. Similarly to the FPSB environment, we observe that using a stronger strategy exploration distribution, instead of simple random search, leads to more true positives for Algorithm 2.

8 DISCUSSION

We introduced a strategic stability property called *probably almost stable*, and showed it can be efficiently verified in large simulation-based games. We presented a hypothesis test for the property and a sequential search method for probably almost stable profiles. We empirically demonstrated the efficacy of our new techniques in the FPSB auction and a cybersecurity game, and we showed how the frequency of true positives can be improved by using simulated annealing as the strategy search method.

We stated the probably almost stable guarantee from the perspective of player 1 in a two-player game, but the concept can be easily extended to n-player games, or to considering all players' deviation probabilities simultaneously.

The probably almost stable concept and associated algorithms provide a general statistical approach to quantify confidence in simulation-based game-theoretic results.

Our probably almost stable concept does not provide any bound on the *regret* of a profile σ, which is the maximum any agent could gain by deviating unilaterally from σ. This limitation is unavoidable by methods that sample payoffs for only a subset of strategies, however, because there could exist a strategy not yet sampled with arbitrarily high payoff for an agent i.

REFERENCES

[1] Alan Agresti and Brent A. Coull. 1998. Approximate is better than "exact" for interval estimation of binomial proportions. *The American Statistician* 52, 2 (1998), 119–126.

[2] Shaunak D. Bopardikar, Alessandro Borri, João P. Hespanha, Maria Prandini, and Maria D. Di Benedetto. 2013. Randomized sampling for large zero-sum games. *Automatica* 49, 5 (2013), 1184–1194.

[3] Lawrence D. Brown, T. Tony Cai, and Anirban DasGupta. 2001. Interval estimation for a binomial proportion. *Statist. Sci.* 16 (2001), 101–117.

[4] Frank Cheng, Junming Liu, Kareem Amin, and Michael P. Wellman. 2016. Strategic payment routing in financial credit networks. In *17th ACM Conference on Economics and Computation*. 721–738.

[5] Charles J. Clopper and Egon S. Pearson. 1934. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika* 26, 4 (1934), 404–413.

[6] Richard Engelbrecht-Wiggans and Elena Katok. 2008. Regret and feedback information in first-price sealed-bid auctions. *Management Science* 54, 4 (2008), 808–819.

[7] Paul W. Goldberg. 2004. Bounds for the convergence rate of randomized local search in a multiplayer load-balancing game. In *23rd ACM Symposium on Principles of Distributed Computing*. 131–140.

[8] Daniel Golovin, Benjamin Solnik, Subhodeep Moitra, Greg Kochanski, John Karro, and D Sculley. 2017. Google Vizier: A service for black-box optimization. In *23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1487–1495.

[9] Patrick R. Jordan, L. Julian Schvartzman, and Michael P. Wellman. 2010. Strategy exploration in empirical games. In *9th International Conference on Autonomous Agents and Multiagent Systems*. 1131–1138.

[10] Richard Klima, Daan Bloembergen, Rahul Savani, Karl Tuyls, Daniel Hennes, and Dario Izzo. 2016. Space debris removal: A game theoretic analysis. *Games* 7 (2016), 20:1–20:18.

[11] Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Julien Perolat, David Silver, Thore Graepel, et al. 2017. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems*. 4193–4206.

[12] Joel Z. Leibo, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. 2017. Multi-agent reinforcement learning in sequential social dilemmas. In *16th International Conference on Autonomous Agents and Multiagent Systems*. 464–473.

[13] Olvi L. Mangasarian. 1964. Equilibrium points of bimatrix games. *J. Soc. Indust. Appl. Math.* 12, 4 (1964), 778–780.

[14] Richard D. McKelvey, Andrew M. McLennan, and Theodore L. Turocy. 2006. Gambit: Software tools for game theory. (2006).

[15] H. Brendan McMahan, Geoffrey J. Gordon, and Avrim Blum. 2003. Planning in the presence of cost functions controlled by an adversary. In *20th International Conference on Machine Learning*. 536–543.

[16] Erik Miehling, Mohammad Rasouli, and Demosthenis Teneketzis. 2015. Optimal defense policies for partially observable spreading processes on Bayesian attack graphs. In *Second ACM Workshop on Moving Target Defense*. 67–76.

[17] Thanh H. Nguyen, Mason Wright, Michael P. Wellman, and Satinder Singh. 2018. Multi-stage attack graph security games: Heuristic strategies, with empirical game-theoretic analysis. *Security and Communication Networks* (2018).

[18] Cynthia Phillips and Laura Painton Swiler. 1998. A graph-based system for network-vulnerability analysis. In *Workshop on New Security Paradigms*. 71–79.

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

[20] L. Julian Schvartzman and Michael P. Wellman. 2009. Stronger CDA strategies through empirical game-theoretic analysis and reinforcement learning. In *8th International Conference on Autonomous Agents and Multiagent Systems*. 249–256.

[21] Ashish Sureka and Peter R. Wurman. 2005. Using tabu best-response search to find pure strategy Nash equilibria in normal form games. In *4th International Conference on Autonomous Agents and Multiagent Systems*. 1023–1029.

[22] Måns Thulin. 2014. The cost of using exact confidence intervals for a binomial proportion. *Electronic Journal of Statistics* 8, 1 (2014), 817–840.

[23] Elaine Wah, Mason Wright, and Michael P. Wellman. 2017. Welfare effects of market making in continuous double auctions. *Journal of Artificial Intelligence Research* 59 (2017), 613–650.