Exponential Lower Bounds on the Double Oracle Algorithm in Zero-Sum Games

Brian Hu Zhang[1] , **Tuomas Sandholm**[1,2,3,4]
[1]Computer Science Department, Carnegie Mellon University
[2]Strategy Robot, Inc.
[3]Strategic Machine, Inc.
[4]Optimized Markets, Inc.
{bhzhang, sandholm}@cs.cmu.edu

Abstract

The double oracle algorithm is a popular method of solving games, because it is able to reduce computing equilibria to computing a series of best responses. However, its theoretical properties are not well understood. In this paper, we provide exponential lower bounds on the performance of the double oracle algorithm in both partially-observable stochastic games (POSGs) and extensive-form games (EFGs). Our results depend on what is assumed about the *tiebreaking scheme*—that is, which meta-Nash equilibrium or best response is chosen, in the event that there are multiple to pick from. In particular, for EFGs, our lower bounds require *adversarial* tiebreaking, whereas for POSGs, our lower bounds apply regardless of how ties are broken.

1 Introduction

The *double oracle algorithm* [McMahan *et al.*, 2003] is a popular practical framework for solving large games. It works by maintaining a *meta-game* comprised of a set of policies for each player, computing a *meta-Nash equilibrium* of the meta-game, and then computing *best responses* to that meta-game and adding those best responses to the meta-game for the next iteration. In essence, it reduces solving *multi-player* games to solving a series of small meta-games, which are easy, and *best-response problems*, which are single-player games. The method (or, more specifically, variations on the deep generalization of it, in which the best responses are replaced with deep RL-based approximate POMDP solvers, commonly referred to as a special case of the *policy-space response oracle* [Lanctot *et al.*, 2017] algorithm), has been successfully applied to large, two-player zero-sum games such as *Barrage Stratego* [McAleer *et al.*, 2020] and *StarCraft* [Vinyals *et al.*, 2019]. In practice, the algorithm tends to converge very fast: even in games far too large to enumerate the state space, only tens or hundreds of iterations are required to reach strong play.

However, to our knowledge, the theoretical properties of double oracle are almost completely unstudied. Indeed, the lack of an efficient convergence guarantee has led to several variants of double oracle being developed which *do* have efficient convergence guarantees, most notably the *sequence-form* [Bošanskỳ *et al.*, 2014] and *extensive-form double oracle* [McAleer *et al.*, 2021] algorithms. In extensive-form games, both of these algorithms are guaranteed to converge in a number of iterations polynomial in the size of the game. Another variant of double oracle, *self-play PSRO* [McAleer *et al.*, 2022] has also been developed that adds *randomized* policies to the meta-game, in the hopes that such policies lead to faster learning. In this paper, however, we focus on the plain version of the double oracle algorithm.[1]

We derive several different partially-observable stochastic games (POSGs) in which double oracle takes exponentially many iterations to converge. The games differ in their structure and in what assumptions need to be made about the choices left unspecified in the algorithm, namely, the choices of *initialization, meta-Nash equilibria*, and *best responses*. For example, if all choices are *random* then we give a *partially-observable stochastic game* with an exponential convergence bound (Theorem 3.2); if all choices can be made *adversarially*, then we give a *tree-form, fully-observable* game (Theorem 3.5). A summary of our results can be found in Table 1.

2 Preliminaries

A *two-player partially-observable stochastic game* (POSG) (hereafter simply *game*) consists of the following elements:[2]

1. A finite *state space* S, *action spaces* A_1, A_2, and *observation space* O with $|O| \leq |S|$;
2. a *starting distribution* $S_0 \in \Delta(S)$;
3. a set of *terminal states* $Z \subset S$;

[1]In multi-player general-sum games, especially when the game is large enough that "best" responses are approximated with deep reinforcement learning, generalizations and variants of the double oracle algorithm have been studied under the name *policy space response oracle* (PSRO) [*e.g.*, Lanctot *et al.* 2017]. In this paper, we adhere to the more traditional name *double oracle* because we are indeed working with the more "standard" two-player version of the algorithm, not any generalization thereof.

[2]The definition used here is more restrictive than many common definitions of POSGs. For example, many authors allow observations to be randomized, or action sets to depend on state, or rewards to be given at nonterminal states and be action-dependent. But since this whole paper concerns only *lower bounds*, adding restrictions makes our results *more* powerful. It also simplifies our notation.

4. for each (s, a_1, a_2) where $s \in S \setminus Z, a_1 \in A_1, a_2 \in A_2$, a probability distribution $p(\cdot|s, a_1, a_2) \in \Delta(S)$ denoting the probability of transitioning to the next state;
5. two *observation function* $o_1, o_2 : S \setminus Z \to O$; and
6. two *reward functions* $R_1, R_2 : Z \to [-1, +1]$ denoting the reward of P1 and P2 respectively, as a function of the terminal state reached.

A game is *zero-sum* if $R_1 = -R_2$. We will make the assumption that the game has a DAG structure: the transition multigraph of the game—that is, the multigraph whose nodes are the states and for which there is an edge (s, s') for each pair $(a_1, a_2) \in A_1 \times A_2$ such that $p(s'|s, a_1, a_2) > 0$—is directed and acyclic. Thus, the terminal states $z \in Z$ are the sinks of this DAG. We will denote the depth of the DAG by k.

A *pure policy* for a player $i \in \{1, 2\}$ is a mapping $\pi_i : O^{\leq d} \to A_i$, where $O^{\leq d}$ denotes the set of sequences on O of length at most k. We denote by Π_i the set of pure policies of player i. A pair of pure policies (π_1, π_2) is a *policy profile* or simply *profile*. A profile induces a distribution over the terminal states Z of the game, given by sampling $s_0 \sim S_0$ and then following (π_1, π_2) until a state $z \in Z$ is reached. We will use $z \sim (\pi_1, \pi_2)$ to denote a sample from this distribution. A *mixed policy* $\mu_i \in \Delta(\Pi_i)$ is a distribution over pure policies. Given mixed profile (μ_1, μ_2), the *expected value* of player i is

$$V_i(\mu_1, \mu_2) = \mathop{\mathbb{E}}_{\substack{\pi_1 \sim \mu_1, \\ \pi_2 \sim \mu_2, \\ z \sim (\pi_1, \pi_2)}} R_i(z).$$

Policy $\pi_i \in \Pi_i$ is a *best response* to a mixed policy μ_{-i} if

$$\pi_i \in \mathop{\mathrm{argmax}}_{\pi_i' \in \Pi_i} V_i(\pi_i', \mu_{-i}).$$

An *ε-Nash equilibrium* is a profile (μ_1, μ_2) such that neither player can improve by more than ε:

$$\max_{\pi_i \in \Pi_i} V_i(\pi_i, \mu_{-i}) - V_i(\mu) \leq \varepsilon.$$

A Nash equilibrium is a 0-Nash equilibrium. In general, computing a Nash equilibrium of a POSG is hard—indeed, even solving POMDPs (*i.e.*, POSGs where $|A_2| = 1$) is PSPACE-complete [Papadimitriou and Tsitsiklis, 1987]. It will be useful to define several special cases of POSGs:

1. A *(fully-observable) stochastic game* is a POSG in which both players observe the true state, *i.e.*, $S = O$ and $o_1(s) = o_2(s) = s$.
2. A *tree-form game* is a POSG in which the transition multigraph is a tree.
3. A *normal-form game* is a stochastic game with a single nonterminal state (which is also the start state). A two-player normal-form game is described by two matrices $V, V_2 \in \mathbb{R}^{A_1 \times A_2}$, where $V_i(a_1, a_2)$ is the reward to player i if P1 plays action a_1 and P2 plays a_2.

Any of the other forms can be converted into normal form at the cost of a larger game: namely, any POSG is equivalent to the normal-form game described by matrices $V_1, V_2 \in \mathbb{R}^{\Pi_1 \times \Pi_2}$. This conversion, however, incurs doubly-exponential blowup in the size of the game in general.

Algorithm 1: The double oracle algorithm. NORMALFORMNASHEQUILIBRIUM returns an exact Nash equilibrium to the normal-form game in which each player picks a policy from its policy set $\widetilde{\Pi}_i$. BESTRESPONSE returns a pure policy that is a best response to the given opponent policy.

Input: POSG, initial strategies $\pi_1^0 \in \Pi_1, \pi_2^0 \in \Pi_2$,
  desired Nash gap $\varepsilon \geq 0$
Output: ε-Nash equilibrium (μ_1, μ_2) of the POSG
$\widetilde{\Pi}_1^0 \leftarrow \{\pi_1^0\}, \widetilde{\Pi}_2^0 \leftarrow \{\pi_2^0\}$
for $t = 1, 2, \ldots$ **do**
  $\mu_1^t, \mu_2^t \leftarrow$ Nash equilibrium of
    normal-form game $(\widetilde{\Pi}_1^{t-1}, \widetilde{\Pi}_2^{t-1})$
  $\pi_1^t \leftarrow$ P1 best response to μ_2^t
  $\pi_2^t \leftarrow$ P2 best response to μ_1^t
  if *Nash gap* $\leq \varepsilon$ **then return** (μ_1^t, μ_2^t)
  $\widetilde{\Pi}_1^t \leftarrow \widetilde{\Pi}_1^{t-1} \cup \{\pi_1^t\}$
  $\widetilde{\Pi}_2^t \leftarrow \widetilde{\Pi}_1^{t-1} \cup \{\pi_2^t\}$

For zero-sum games, in each of the special cases, there are polynomial-time algorithms for exactly computing a Nash equilibrium: in the fully-observable case, one can perform backwards induction (value iteration) starting from the leaves, solving each state via a linear program; tree-form POSGs are a subclass of *extensive-form games*, and Koller *et al.* [1994] describe an LP-based method that runs in polynomial time.

2.1 The Double Oracle Algorithm

Pseudocode for the double oracle algorithm is given in Algorithm 1. The algorithm is simple: it iteratively maintains a *meta-game* $(\widetilde{\Pi}_1, \widetilde{\Pi}_2)$, computes a *meta-Nash equilibrium* (μ_1, μ_2) to that meta-game, computes best responses (π_1, π_2) in the full game, and adds those best responses to the meta-game. Double oracle clearly converges in a finite number of steps: there are only a finite number of pure policies, and each iteration of the main loop must add a pure policy to at least one player's meta-game policy set (if both best responses π_1, π_2 are already in the policy sets, then the Nash gap would be 0).

The meta-game on iteration t is a $t \times t$ normal-form game. For zero-sum games at least, as specified above, Nash equilibria can be easily computed in polynomial time via linear programming [von Neumann, 1928]. Thus, the entire complexity of Algorithm 1 lies in the best responses (which are POMDPs) and the number of iterations t until the algorithm terminates. For nonzero-sum games, Nash equilibrium computation is in general hard [Chen *et al.*, 2009]. However, we will ignore these computational issues and focus our attention on the number of iterations it takes for double oracle to converge.

The double oracle algorithm is not affected by the game representation. For example, running double oracle on a POSG and running double oracle on the normal form of that POSG would produce the same result. Therefore, for the rest of the paper, we will call two games *(strategically) equivalent* if they induce the same normal form.

3 Main Results

As suggested above, the main results in this paper are *lower bounds* on the complexity of the double oracle algorithm. In particular, we will give several game examples in which double oracle, under various assumptions about the best response oracle, fails to converge to an ε-equilibrium, for moderately-sized ε, until t is exponentially large.

In general, a stochastic game may have no Nash equilibria with small support. For example, consider the k-bit "generalized matching pennies" game in which P1 picks a string $\pi_1 \in \{0,1\}^k$ one bit at a time, and P2 simultaneously attempts to guess that string, also one bit at a time, with P2 winning if and only if P1 and P2 guess the same string.

This game for $k = 4$ is depicted in Figure 1. Intuitively, it is nothing more than a finite automaton that reads two bitstrings $a_1, a_2 \in \{0,1\}^k$ (interpreted as natural numbers in $\{0, 1, \ldots, 2^k-1\}$) in parallel, and outputs the reward $u(a_1, a_2)$ as specified by the normal-form game: that is, it returns -1 if the strings are equal and $+1$ otherwise. This proves:

Theorem 3.1. *For every $k \geq 1$, there exists a zero-sum fully-observable stochastic game with $O(k)$ nodes in which, regardless of initialization, meta-Nash, or best responses, double oracle takes $2^{\Theta(k)}$ iterations to find an exact equilibrium.*

However, the "generalized matching pennies" game is not ideal as a counterexample, for multiple reasons:

1. *Polynomial-time approximation*: While double oracle fails to converge to *exact* equilibrium in polynomially many iterations, it will converge to a ε-equilibrium in $O(1/\varepsilon)$ iterations: one can check inductively that, at odd iterations, P1 will add an arbitrary new policy to its support $\widetilde{\Pi}_1$, and P2 will add the same policy at the next (even) iteration. Thus, after $2t$ iterations we will have $\widetilde{\Pi}_1^{2t} = \widetilde{\Pi}_2^{2t}$ and (μ_1^{2t}, μ_2^{2t}) will be a $1/t$-equilibrium. This is still a reasonable convergence rate.
2. *High support*. As mentioned above, the game has only high-support equilibria.

The main counterexamples in our paper will fix both of these issues. In particular, all our counterexamples will be families of games in which *there is a Nash equilibrium with constant support size*, and yet double oracle *fails to find any ε-approximate equilibrium* in $\mathrm{poly}(N, 1/\varepsilon)$ iterations, where N is the size of the representation of the POSG. These counterexamples are summarized in Table 1.

Theorem 3.2. *For every $k \geq 1$, there exists a zero-sum POSG with $O(k)$ states and a pure Nash equilibrium in which, in the double oracle algorithm,*

- *the meta-Nash equilibria and the best responses are unique on every iteration, and*
- *for ε constant, if the starting policies π_1^0, π_2^0 are chosen uniformly at random[3], then double oracle takes $\Theta(2^k)$ iterations in expectation.*

[3]Choosing starting policies at random means choosing a *pure* policy π_1^0 from Π_1 uniformly at random, not setting π_1^0 to be the uniformly random policy.

Proof. The proof is based on a simple normal-form game that we call the *n-bigger-number game*. In the n-bigger-number game, each player's action space is $A_1 = A_2 = [n] := \{0, \ldots, n-1\}$, and the rules are as follows. Both players simultaneously select numbers $a_i \in [n]$. If $a_i = a_j$, then both players score 0. Otherwise, the player who plays the bigger number scores 1, unless $|a_i - a_j| = 1$ in which case they score 2.

We first analyze the behavior of double oracle in the n-bigger-number game with random starting policies. For each t, let $M(t)$ be the largest number in the support of either player's policy set Π_i^t. Then with constant probability, $M(0) \leq n/2$. Further, μ_1^t is supported on $\{0, \ldots, M(t-1)\}$. Then the best response π_2^t to μ_1^t is at most $M(t-1)+1$, because any number larger than t performs worse than $\max \mathrm{supp}(\mu_1^t) + 1 \leq t$. Thus, $M(t) \leq M(t-1) + 1$ for all $t \geq 1$. Equilibrium can only be reached when $M(t) = n$, because the only equilibrium of the game is (n, n). Therefore, with constant probability, double oracle takes $\Theta(n)$ iterations, and therefore the expected number of iterations for double oracle is also $\Theta(n)$.

We now show that the n-bigger-number game, for $n = 2^k$, is equivalent to a POSG with $O(k)$ nodes, which would complete the proof. Consider the POSG depicted in Figure 2 (for $k = 4$, easily generalizable). Like Figure 1, this POSG is essentially a finite automaton that reads two bitstrings a_1, a_2 simultaneously, and outputs the required value. The reward depends on the value of $a_1 - a_2$, in particular, whether it is greater than 1, equal to 1, equal to 0, equal to -1, or less than -1. The center row of nodes captures the states in which the substrings read are currently equal (If that continues until the last timestep, then the numbers are equal). The row above the center captures the states in which $a_1 \neq a_2$ but it is still possible for $a_1 = a_2 + 1$. (This happens if $a_1 = x10^\ell$ and $a_2 = x01^\ell$ for some string x and integer ℓ.) The row below the center is the same but with the players flipped.

Since observations are trivial, a pure policy in this POSG is specified by a vector $\pi_i \in \{0,1\}^k$, whose jth index specifies the action played by player i at time $j \in [k]$. The vector π_i is then identified with the pure action in the 2^k-bigger-number game whose binary representation is π_i. This POSG is equivalent to the 2^k-bigger-number game. □

The next two results will be similar to the above result, but will have increasingly stringent requirements on the structure of the game—first, stochastic games, and then tree-form stochastic games. In exchange, we will also need more stringent requirements on the behavior of the double oracle algorithm. In particular, the meta-Nash equilibria and best responses used by double oracle may no longer be unique, so we will need to make assumptions on how they are chosen. Whenever the choice is not unique, we will always assume *adversarial* choices for the algorithm—that is, we will assume that meta-Nash equilibria and best responses are chosen to make double oracle run for as long as possible.

Theorem 3.3. *For every $k \geq 1$, there exists a zero-sum* fully-*observable stocastic game with $O(k)$ states and a pure Nash equilibrium , in which, in the double oracle algorithm,*

- *the meta-Nash equilibria are unique on every iteration,*

	game properties			double oracle assumptions					
	ZS	FO	TF	Nash support	initialization	meta-Nash	best responses	$\lvert S\rvert$	ε^*
Theorem 3.1	✓	✓	✗	$2^{\Theta(k)}$	—	—	—	$O(k)$	$2^{-\Theta(k)}$
Theorem 3.2	✓	✗	✗	1	random	—	—	$O(k)$	$\Theta(1)$
Theorem 3.3	✓	✓	✗	1	random	—	adversarial	$O(k)$	$\Theta(1)$
Theorem 3.4	✗	✗	✓	1	adversarial	adversarial	—	$\text{poly}(k)$	$\Theta(1/k)$
Theorem 3.5	✓	✓	✓	2	adversarial	adversarial	adversarial	$O(k)$	$\Theta(1/k)$

Table 1: Summary of main results. *Nash support* gives the minimum support per player, in pure policies, of any exact Nash equilibrium. In all cases double oracle takes $2^{\Theta(k)}$ iterations to converge to an ε-equilibrium for every $\varepsilon < \varepsilon^*$. 'ZS', 'FO', and 'TF' mean zero-sum, fully-observable, and tree-form, respectively.

- *the best responses are* not *unique on every iteration, and*
- *for $\varepsilon < 2$, if the starting policies π_1^0, π_2^0 are chosen uniformly at random, double oracle* with adversarial best responses *takes $2^k - 1$ iterations.*

Proof. We will define a *n-weak bigger-number game* similar to the n-bigger-number game used in the proof of Theorem 3.2. In the *weak n-bigger-number game*, two players simultaneously select a number $a_i \in [n]$, and whoever picks the bigger number wins (scores 1).

Unlike the bigger-number game, best responses will not be unique in the weak bigger-number game. For example, every number bigger than 0 is a best response to 0. However, we can still replicate the behavior of double oracle on the bigger-number game, because the same conditions for that behavior still hold: namely, the only Nash equilibrium is (n, n), and $\max \text{supp}(\mu_1^t) + 1$ is always a best response to μ_1^t. Therefore, if we always adversarially choose this best response, an identical analysis holds, and the expected runtime of double oracle is $\Theta(n)$ iterations.

We now only need to represent the 2^k-weak bigger-number game as a stochastic game. Consider the stochastic game in Figure 3, which is this time a *fully-observable* game[4]. Once again, this POSG is essentially a finite automaton that computes the game value. As before, we relate the policies, which are vectors $\pi_i \in \{0,1\}^k$, to numbers in $\{0, \dots, 2^k - 1\}$ via their binary representation, and from this it is easy to see that the normal form of this stochastic game is indeed the 2^k-weak bigger-number game. □

Our next result is the only result that uses a *nonzero-sum game*, and the first of two results concerning *tree-form* games.

Theorem 3.4. *For every $k \geq 1$, there exists a* nonzero-sum, tree-form, partially-observable *stochastic game with* $\text{poly}(k)$ *states, and a pure Nash equilibrium, in which, for $\varepsilon < 1/k$, there exist starting policies π_1^0, π_2^0 such that double oracle with adversarial meta-Nash equilibria takes $\Theta(2^k)$ iterations to converge.*

Proof. As before, we define the normal-form game first. In the *n-incrementing game*, two players simultaneously pick numbers $a_i \in [n]$. If $a_i = a_j + 1$ then player i scores α and player j scores $-\beta$, where $\beta > \alpha > 0$. If $a_i = a_j$ then both players score 0. Otherwise both players score a negative number.

It is easy to see that, in the subgame where both players are restricted to $\{0, \dots, t\} \subseteq [n]$, (t, t) is a Nash equilibrium (in fact, the unique welfare-maximizing equilibrium) and $t + 1$ is a best response for both players. Thus, if both players are initialized at $\widetilde{\Pi}_i^0 = \{0\}$, convergence will only happen after will only converge after n iterations. We will set $n = 2^k$, and show that this game is representable as a stochastic game with $\text{poly}(k)$ states.

Consider the stochastic game defined as follows. Both players have action sets of size $2k$, identified with bitstrings consisting of completely repeated digits, *i.e.*, 0, 1, 00, 11, 000, 111, *etc.* For cleanliness we will write 0^ℓ to be the string with 0 repeated ℓ times, and 1^ℓ for the string with 1 repeated ℓ times. These strings will denote the *trailing runs* of the players' bit strings. The transitions are as follows. At the root state, if both players play the same bit and different lengths, then both players score -2. If the players play different-length strings and neither player has played a string of length 1, both players score -2. Otherwise, the game continues.

At this point, there are three possibilities. From here onwards, players are forced to play either 0 or 1: any other action immediately terminates the game with both players scoring -2 (and is therefore dominated).

1. Both players have played 0^ℓ or 1^ℓ. In this case, bit $i \in \{1, \dots, k - \ell - 1\}$ is drawn uniformly at random and disclosed to both players, and both players choose an action. Both players score 0 if the bits match, and -1 otherwise.
2. One player has played 0^ℓ, and the other has player 1^ℓ. In this case, a bit $i \in \{1, \dots, k - \ell - 1\}$ is drawn uniformly at random and disclosed to both players, and both players then choose an action again. The player who played 1^ℓ scores -1. The player who played 0^ℓ scores $1/2k$ if the bits match, and -1 otherwise.
3. One player (WLOG, P1) has played 0^ℓ (for $\ell > 1$), and the other has played 1. In this case, a bit $i \in \{1, \dots, k - 2\}$ is selected at random. Then both players select an action. If $i = k - \ell$ then P1 is forced to play 1; if $i > k - \ell$ then P1 is forced to play 0. P1 scores -1. P2 scores $1/2k$ if the bits match, and -1 otherwise.
4. One player (WLOG, P1) has played 1^ℓ (for $\ell > 1$), and the other has played 0. In this case, a bit $i \in \{1, \dots, k -$

[4]The observations in this game are actually irrelevant, because there is only one possible state corresponding to each history length.

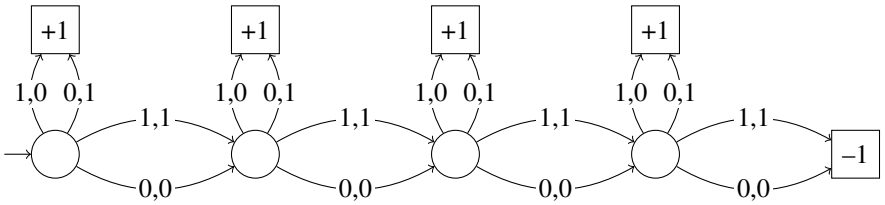

Figure 1: The k-bit guess-the-string game, here depicted for $k = 4$. The action spaces are $A_1 = A_2 = \{0, 1\}$. The start state is the leftmost state, labeled with $\rightarrow$. Terminal states are drawn as rectangles, and their rewards are written within them. Transitions are deterministic, and edges are labeled with the transitions that take them there.

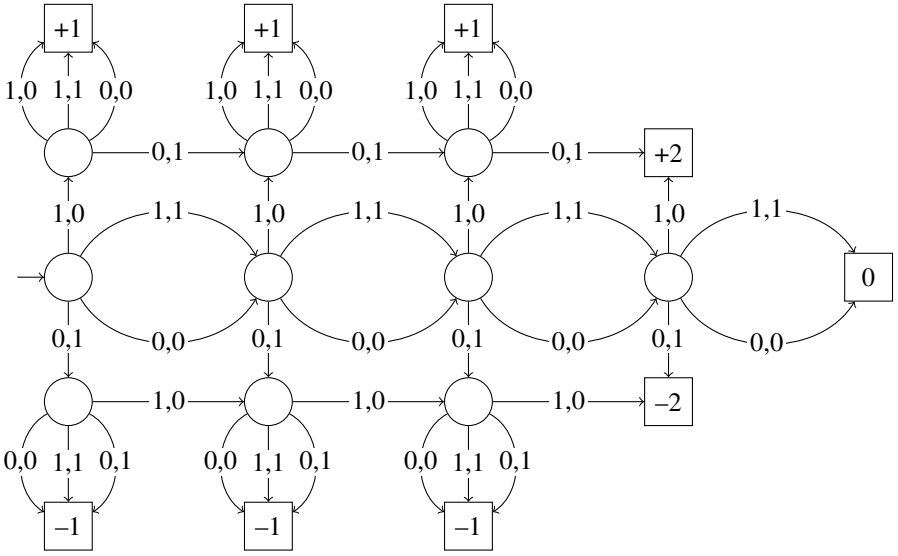

Figure 2: The 2^k-bigger-number game used in Theorem 3.2, here depicted for $k = 4$. Observations are trivial: $|O| = 1$.

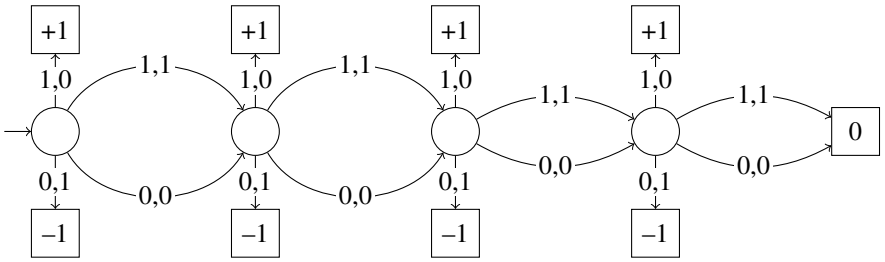

Figure 3: The 2^k-weak bigger-number game used in Theorem 3.3, here depicted for $k = 4$.

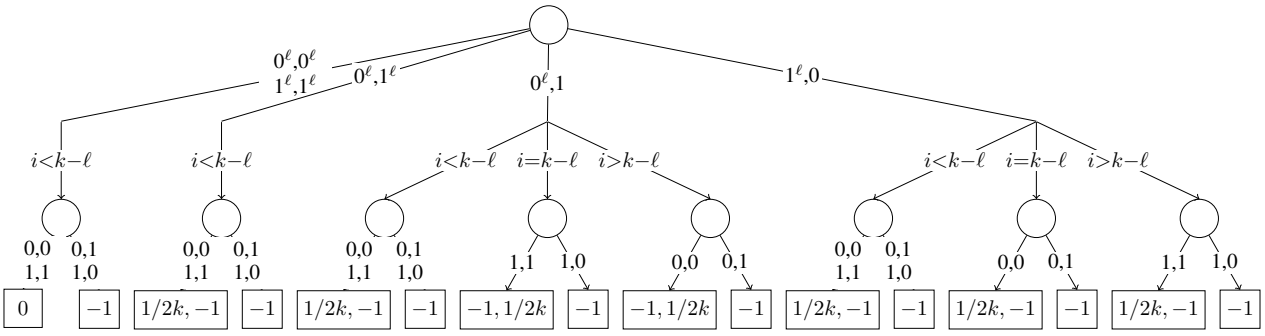

Figure 4: A depiction of the game used in Theorem 3.4. Observations are not shown: the only nontrivial observation each player makes is the randomly-selected index i. Not all actions and transitions are shown. If a terminal node contains only one reward, then that is the reward of both players.

$2\}$ is selected at random. Then both players select an action. If $i = k - \ell$ then P1 is forced to play 0. If $i > k - \ell$ then P1 is forced to play 1. P2 scores -1. P1 scores $1/2k$ if the bits match, and -1 otherwise.

A sketch of the game is depicted in Figure 4. Like the previous three proofs, we still essentially want a state machine to discriminate between the same five classes ($a_1 - a_2 > 1, = 1, = 0, = -1, < -1$) but now we need the game to be tree-form. Since the comparison between the numbers requires knowing how long the trailing run of ones (or zeros) is, we ask the players for this information up-front—that is, both players at the start state choose the trailing runs of their numbers from the set $\{0, 1, 00, 11, 000, 111, \dots\}$ of size $2k$. Conditioned on these choices, the reward function is linear in the prefixes of the two players' bitstrings, and hence it can be represented by a single layer of the game tree in which a bit is selected at random and then the players pick assignments to that bit.

An undominated pure policy (for either player) consists of a trailing run 0^ℓ or 1^ℓ, and assignments to each bit $i \in \{1, \dots, k - \ell - 1\}$. Thus, such strategies correspond exactly to the bitstrings in $\{0, 1\}^n$. It is easy to check that the utilities in the game restricted to undominated strategies satisfy the conditions of the n-incrementing game, completing the proof. □

Our final result will involve a case where both the meta-Nash equilibria *and* the best responses are not unique, and therefore we will assume that both are adversarially chosen. However, the game in the counterexample will have the most stringent structure: the counterexample is a *zero-sum tree-form, fully-observable stochastic game*.

Theorem 3.5. *For every $k \geq 1$, there exists a zero-sum fully-observable,* tree-form *stochastic game with $O(k)$ states and a Nash equilibrium of support size 2 for each player in which, for $\varepsilon < 2/k$, there exist starting policies π_1^0, π_2^0 such that double oracle with adversarial meta-Nash equilibria and best responses takes at least 2^{k-1} iterations.*

Proof. Unlike in the previous two proofs, in this proof it will be most convenient to start by defining the stochastic game without first discussing its normal form. Consider the following game. There are k nonterminal states, $s_1, \dots, s_k$. The starting distribution S_0 is uniform on $\{s_1, \dots, s_k\}$. At each state, the players will each play a single action $a_i \in \{0, 1\}$, and then the game will end. It remains only to define the rewards.

- At state s_1, P2 wins if and only if the players did not play the same action. That is, s_1 is a matching pennies game.
- At state s_j for $j > 1$, P2 wins if and only if P1 played 0 and P2 played 1.

The winner gets value $+1$, and the loser gets value -1.

The equilibrium value of this game is $1 - 1/k$ for P1: the profile "play uniform random at s_1 and 1 at all other states" is an equilibrium policy for both players of support size 2. As before, we will identify pure strategies $\pi_i \in \{0, 1\}^k$ with the numbers they encode in binary. In this notation, let $\pi_1^0 = 2^k - 1$ and $\pi_2^0 = 0$. Then we will show that, for $t \in \{1, \dots, 2^{k-1} - 1\}$, the following adversarial choices of meta-Nash and best responses are possible in the double oracle algorithm:

1. $t - 1$ is a best response for P1 against P2 playing $t - 1$,
2. t is a best response for P2 against P1 playing $2^k - 1$,
3. $\widetilde{\Pi}_1^t = \{2^k - 1\} \cup \{0, \dots, t - 1\}$, and $\widetilde{\Pi}_2^t = \{0, \dots, t\}$, and
4. $(2^k - 1, t - 1)$ is a meta-Nash equilibrium if $(\widetilde{\Pi}_1^t, \widetilde{\Pi}_2^t)$, that has equilibrium gap $2/k$ in the full game,

We now prove all four points above by induction.

1. For P1, playing $t - 1$ against $t - 1$ wins all states, so it is a best response.
2. Against $2^k - 1$, P2 can only win the matching pennies game, which P2 does by playing any policy in the range $[0, 2^{k-1} - 1]$. t is indeed such a policy.
3. This follows from the previous two points and the definition of the double oracle algorithm.
4. The profile $(2^k - 1, t - 1)$ scores $1 - 2/k$ for P1 since P1 loses the matching pennies game but wins all others by playing 1. P2 cannot improve upon this. P1 can only improve by winning at all states, but in order to do that, P1 must play a policu in the range $[t-1, 2^{k-1}-1]$. However, P1's policy set $\widetilde{\Pi}_1^{t-1}$ only contains $\{0, \dots, t - 2\}$ by induction hypothesis, so P1 cannot win all states, and therefore $(2^k - 1, t - 1)$ is a meta-Nash equilibrium.

This completes the induction and therefore the proof, since with these choices, the Nash gap computed by double oracle will stay at $2/k$ until at least iteration 2^{k-1}. □

4 Discussion and Related Work

In this section, we discuss a few alternative algorithms similar to the double oracle algorithm, and how they relate to the results in this paper.

4.1 Fictitious Play

Another common algorithm for reducing multi-player to single-player games is *fictitious play*. Fictitious play differs from double oracle only in the choice of opponent policies μ_{-i}^t against which player i computes the best response π_i^t. While double oracle uses a Nash equilibrium of the restricted game defined by the policies already discovered, fictitious play uses a simple uniform average over those policies:

$$\mu_{-i}^t := \frac{1}{t} \sum_{\tau=0}^{t-1} \pi_{-i}^{(\tau)}.$$

Although this change seems simple, the two algorithms behave very differently in theory. For example, double oracle is guaranteed to converge in at most $|\Pi|$ iterations, where Π is the set of policies, since at least one policy is added on every iteration until convergence is reached. However, proving (or disproving) a $\mathrm{poly}(|\Pi|, 1/\varepsilon)$-time convergence rate for fictitious play, even in zero-sum games is one of the oldest open problems in game theory, known as *Karlin's conjecture* [Karlin, 1959]. Similarly to our discoveries, however, the behavior

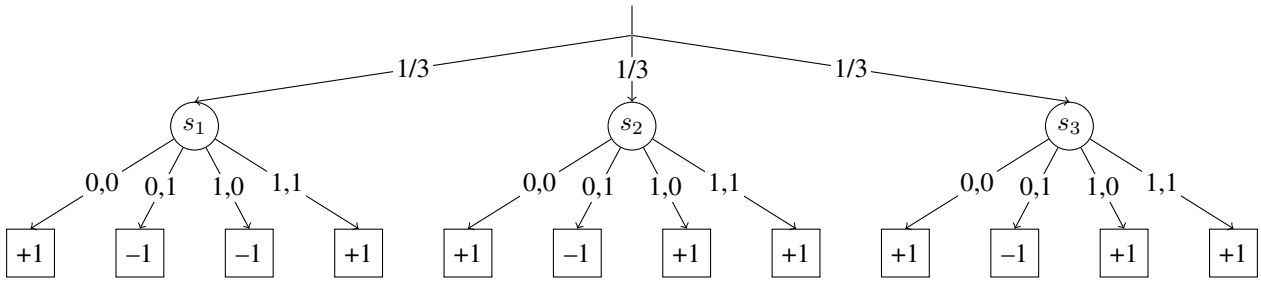

Figure 5: A depiction of the game used in Theorem 3.5, for $k = 3$. Edges to the start states are labeled with their starting probabilities (1/3).

of fictitious play is known to depend on assumptions about tiebreaking. In particular, it is known that for normal-form games whose payoff matrix is diagonal, the convergence rate of fictitious play is polynomial if the best responses are chosen using a consistent tiebreaking method [Abernethy *et al.*, 2021], but not if they are chosen adversarially [Daskalakis and Pan, 2014].

4.2 α-Best Response Dynamics and Potential Games

In *best response dynamics*, we simply set $\mu_{-i}^t = \pi_{-i}^{t-1}$. That is, each player simply best responds to the opponent's previous policy. In zero-sum games, best response dynamics usually will not converge to equilibria: indeed, since π^t is always pure, best response dynamics cannot converge whenever there is no pure equilibrium. However, best response dynamics have been considered in the class of *potential games*, which are, roughly speaking, games that "look like" ones in which every player has the same utility function. In this class of games, it has been observed [Awerbuch *et al.*, 2008; Chien and Sinclair, 2011] that it is sometimes better to *limit* players to only playing best responses if they improve the player's utility by more than some parameter α.

One may ask whether a similar change affects our lower bounds. That is, suppose that, in the double oracle algorithm, the best response π_i^t is only added to Π_i^t if $V(\pi_i^t, \mu_{-i}^t) - V(\mu^t) \geq \alpha$, where $\varepsilon \geq \alpha > 0$. Let us call this algorithm *α-double oracle*.

- In Theorem 3.1, the best response of P1 at iteration $2t$ improves the value by $2/t$, and the best response of P2 at iteration $2t + 1$ improves the value by a full 2. Thus, the theorem is unaffected.
- In Theorem 3.2, Theorem 3.3, and Theorem 3.4, the value improvement of every player on every iteration is equal to the Nash gap. Therefore, these results are unaffected.
- Theorem 3.5 *is* affected. That result relies on the ability for P2 to add the best response $\pi_2^t = t$, which does not improve P2's value at all. Thus, the result breaks for every $\alpha > 0$.

5 Conclusions and Future Research

We have shown, to our knowledge, the first exponential lower bounds on the convergence time (in number of iterations) of the double oracle algorithm. We leave several natural questions for future research.

- Can the gaps in Table 1 be closed? For example, does there exist a tree-form POSG in which the double oracle algorithm must take exponentially many iterations with any of the adversarial assumptions removed? Does there exist a fully-observable stochastic game in which the double oracle algorithm is exponential even with non-adversarial best responses?
- Are there "simple" modifications to double oracle, for example, α-double oracle as suggested in Section 4.2, that guarantee polynomial worst-case bounds in certain cases (*e.g.*, zero-sum tree-form games)?

Acknowledgements

This work is supported by the Vannevar Bush Faculty Fellowship ONR N00014-23-1-2876, National Science Foundation grants RI-2312342 and RI-1901403, ARO award W911NF2210266, and NIH award A240108S001. Brian Hu Zhang's work is supported in part by the CMU Computer Science Department Hans Berliner PhD Student Fellowship.

References

[Abernethy *et al.*, 2021] Jacob Abernethy, Kevin A Lai, and Andre Wibisono. Fast convergence of fictitious play for diagonal payoff matrices. In *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1387–1404. SIAM, 2021.

[Awerbuch *et al.*, 2008] Baruch Awerbuch, Yossi Azar, Amir Epstein, Vahab Seyed Mirrokni, and Alexander Skopalik. Fast convergence to nearly optimal solutions in potential games. In *ACM Conference on Economics and Computation (EC)*, pages 264–273, 2008.

[Bošanskỳ *et al.*, 2014] Branislav Bošanskỳ, Christopher Kiekintveld, V Lisý, and Michal Pěchouček. An exact double-oracle algorithm for zero-sum extensive-form games with imperfect information. *Journal of Artificial Intelligence Research*, pages 829–866, 2014.

[Chen *et al.*, 2009] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM*, 2009.

[Chien and Sinclair, 2011] Steve Chien and Alistair Sinclair. Convergence to approximate nash equilibria in congestion games. *Games and Economic Behavior*, 71(2):315–327, 2011.

[Daskalakis and Pan, 2014] Constantinos Daskalakis and Qinxuan Pan. A counter-example to karlin's strong conjecture for fictitious play. In *Annual Symposium on Foundations of Computer Science (FOCS)*, pages 11–20. IEEE, 2014.

[Karlin, 1959] SAMUEL Karlin. Mathematical methods and theory in games, programming and economics. 1 addisonwesley. *Reading, Mass*, 1959.

[Koller *et al.*, 1994] Daphne Koller, Nimrod Megiddo, and Bernhard von Stengel. Fast algorithms for finding randomized strategies in game trees. In *ACM Symposium on Theory of Computing (STOC)*, 1994.

[Lanctot *et al.*, 2017] Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 4190–4203, 2017.

[McAleer *et al.*, 2020] Stephen McAleer, John B Lanier, Roy Fox, and Pierre Baldi. Pipeline PSRO: A scalable approach for finding approximate Nash equilibria in large games. *Conference on Neural Information Processing Systems (NeurIPS)*, 33:20238–20248, 2020.

[McAleer *et al.*, 2021] Stephen McAleer, Kevin A Wang, Pierre Baldi, and Roy Fox. Xdo: A double oracle algorithm for extensive-form games. *Conference on Neural Information Processing Systems (NeurIPS)*, 34:23128–23139, 2021.

[McAleer *et al.*, 2022] Stephen McAleer, John Banister Lanier, Kevin Wang, Pierre Baldi, Roy Fox, and Tuomas Sandholm. Self-play PSRO: Toward optimal populations in two-player zero-sum games. *arXiv preprint arXiv:2207.06541*, 2022.

[McMahan *et al.*, 2003] H Brendan McMahan, Geoffrey J Gordon, and Avrim Blum. Planning in the presence of cost functions controlled by an adversary. In *International Conference on Machine Learning (ICML)*, pages 536–543, 2003.

[Papadimitriou and Tsitsiklis, 1987] Christos H Papadimitriou and John N Tsitsiklis. The complexity of markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

[Vinyals *et al.*, 2019] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[von Neumann, 1928] John von Neumann. Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, 100:295–320, 1928.