Anytime Optimal PSRO for Two-Player Zero-Sum Games

Stephen McAleer[123], Kevin Wang[1], John Lanier[1], Marc Lanctot[2], Pierre Baldi[1], Tuomas Sandholm[3], Roy Fox[1]

[1]University of California, Irvine Department of Computer Science
[2]DeepMind [3]Carnegie Mellon University
Correspondence to Stephen McAleer at mcaleer.stephen@gmail.com

Abstract

Policy space response oracles (PSRO) is a multi-agent reinforcement learning algorithm that has achieved state-of-the-art performance in very large two-player zero-sum games. PSRO is based on the tabular *double oracle (DO)* method, an algorithm that is guaranteed to converge to a Nash equilibrium, but may increase exploitability from one iteration to the next. We propose *anytime double oracle (ADO)*, a tabular double oracle algorithm for 2-player zero-sum games that is guaranteed to converge to a Nash equilibrium while decreasing exploitability from one iteration to the next. Unlike DO, in which the restricted distribution is based on the restricted game formed by each player's strategy sets, ADO finds the restricted distribution for each player that minimizes its exploitability against any policy in the full, unrestricted game. We also propose a method of finding this restricted distribution via a no-regret algorithm updated against best responses, called *RM-BR DO*. Finally, we propose *anytime PSRO (APSRO)*, a version of ADO that calculates best responses via reinforcement learning. In experiments on Leduc poker and random normal form games, we show that our methods achieve far lower exploitability than DO and PSRO and decrease exploitability monotonically.

Introduction

Policy Space Response Oracles is a multi-agent reinforcement learning (RL) method that is based on the tabular double oracle (DO) algorithm (McMahan, Gordon, and Blum 2003) for finding an approximate *Nash equilibrium (NE)* in two-player zero-sum games. Methods based on PSRO have achieved state-of-the-art performance on large imperfect-information two-player zero-sum games such as Starcraft (Vinyals et al. 2019) and Stratego (McAleer et al. 2020).

In PSRO, each player maintains, and adds to, a population of policies. PSRO can be run until convergence, in which case it outputs a restricted distribution over population policies which corresponds to a NE solution to the original game. In practice, however, PSRO is terminated early in large games. This can be a problem because the PSRO restricted distribution over the population policies is not guaranteed to decrease in exploitability every iteration. As a re-

sult, if PSRO is terminated early, the final restricted distribution could even be more exploitable than the initial one.

In this paper, we propose a new PSRO variant that, in each iteration, finds the least-exploitable restricted distribution over the population policies of each player. We present two versions of our algorithm: an exact version (called *anytime double oracle (ADO)*), and an RL version (called *anytime PSRO (APSRO)*).

Anytime double oracle (ADO) is a modification of *range of skill (ROS)* (Zinkevich, Bowling, and Burch 2007) that finds a restricted Nash equilibrium over two restricted games: one per player. Each player's restricted game is defined such that their strategies are restricted to be within their population, but the opponent is unrestricted. ADO adds a best response to each player's restricted distribution to the other player's population. ADO is guaranteed to not increase exploitability from one iteration to the next and is guaranteed to converge to a Nash equilibrium in a number of iterations at most equal to the number of pure strategies in the game.

Next, we show that an existing method, *regret minimization against a best response (RM-BR)* (Johanson et al. 2012), can be used to calculate the least exploitable restricted distribution in a double oracle algorithm via regret minimization against a best response. This algorithm, which we refer to as RM-BR DO, can efficiently compute a least-exploitable restricted distribution because without solving the large restricted game. However, in large games where computing best responses via RL is necessary, both of these methods becomes prohibitively expensive.

Anytime Policy Space Response Oracles (APSRO) solves this problem by updating the restricted distribution using a no-regret algorithm trained against a single best response that is itself being continually trained via reinforcement learning against the restricted distribution. Although this method lacks theoretical guarantees enjoyed by the exact methods, we find that in practice it can perform much better than PSRO, and tends to not increase exploitability. To the best of our knowledge, ADO is the first double oracle method that both (1) is proven to converge to a Nash equilibrium in a number of iterations bounded by the number of pure strategies of the game; and (2) does not increase exploitability from one iteration to the next. APSRO scales up ADO to large games using reinforcement learning.

To summarize, our contributions are as follows:

- We present ADO, a double oracle algorithm that is guaranteed to not increase exploitability from one iteration to the next.

- We present RM-BR DO, a method for efficiently finding the least-exploitable restricted distribution in ADO via a no-regret procedure against best responses.

- We present APSRO, a reinforcement learning version of ADO that outperforms PSRO in all of our experiments and, unlike PSRO, tends to not increase exploitability.

Background

We consider extensive-form games with perfect recall (Hansen, Bernstein, and Zilberstein 2004). An extensive-form game progresses through a sequence of player actions, and has a *world state* $w \in \mathcal{W}$ at each step. In an N-player game, $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$ is the space of joint actions for the players. $\mathcal{A}_i(w) \subseteq \mathcal{A}_i$ denotes the set of legal actions for player $i \in \mathcal{N} = \{1, \ldots, N\}$ at world state w and $a = (a_1, \ldots, a_N) \in \mathcal{A}$ denotes a joint action. At each world state, after the players choose a joint action, a transition function $\mathcal{T}(w, a) \in \Delta^{\mathcal{W}}$ determines the probability distribution of the next world state w'. Upon transition from world state w to w' via joint action a, player i makes an *observation* $o_i = \mathcal{O}_i(w, a, w')$. In each world state w, player i receives a reward $\mathcal{R}_i(w)$. The game ends when the players reach a terminal world state. In this paper, we consider games that are guaranteed to end in a finite number of actions.

A *history* is a sequence of actions and world states, denoted $h = (w^0, a^0, w^1, a^1, \ldots, w^t)$, where w^0 is the known initial world state of the game. $\mathcal{R}_i(h)$ and $\mathcal{A}_i(h)$ are, respectively, the reward and set of legal actions for player i in the last world state of a history h. An *information set* for player i, denoted by s_i, is a sequence of that player's observations and actions up until that time $s_i(h) = (a_i^0, o_i^1, a_i^1, \ldots, o_i^t)$. Define the set of all information sets for player i to be $\mathcal{I}_i$. The set of histories that correspond to an information set s_i is denoted $\mathcal{H}(s_i) = \{h : s_i(h) = s_i\}$, and it is assumed that they all share the same set of legal actions $\mathcal{A}_i(s_i(h)) = \mathcal{A}_i(h)$.

A player's *strategy* π_i is a function mapping from an information set to a probability distribution over actions. A *strategy profile* π is a tuple $(\pi_1, \ldots, \pi_N)$. All players other than i are denoted $-i$, and their strategies are jointly denoted π_{-i}. A strategy for a history h is denoted $\pi_i(h) = \pi_i(s_i(h))$ and $\pi(h)$ is the corresponding strategy profile. When a strategy π_i is learned through RL, we refer to the learned strategy as a *policy*.

The *expected value (EV)* $v_i^\pi(h)$ for player i is the expected sum of future rewards for player i in history h, when all players play strategy profile π. The EV for an information set s_i is denoted $v_i^\pi(s_i)$ and the EV for the entire game is denoted $v_i(\pi)$. A *two-player zero-sum* game has $v_1(\pi) + v_2(\pi) = 0$ for all strategy profiles π. The EV for an action in an information set is denoted $v_i^\pi(s_i, a_i)$. A *Nash equilibrium (NE)* is a strategy profile such that, if all players played their NE strategy, no player could achieve higher EV by deviating from it. Formally, π^* is a NE if $v_i(\pi^*) = \max_{\pi_i} v_i(\pi_i, \pi_{-i}^*)$ for each player i.

The *exploitability* $e(\pi)$ of a strategy profile π is defined as $e(\pi) = \sum_{i \in \mathcal{N}} \max_{\pi_i'} v_i(\pi_i', \pi_{-i})$. A *best response (BR)* strategy $\mathbb{BR}_i(\pi_{-i})$ for player i to a strategy π_{-i} is a strategy that maximally exploits π_{-i}: $\mathbb{BR}_i(\pi_{-i}) = \arg\max_{\pi_i} v_i(\pi_i, \pi_{-i})$. An *$\epsilon$-best response ($\epsilon$-BR)* strategy $\mathbb{BR}_i^\epsilon(\pi_{-i})$ for player i to a strategy π_{-i} is a strategy that is at most ϵ worse for player i than the best response: $v_i(\mathbb{BR}_i^\epsilon(\pi_{-i}), \pi_{-i}) \geq v_i(\mathbb{BR}_i(\pi_{-i}), \pi_{-i}) - \epsilon$. An *$\epsilon$-Nash equilibrium ($\epsilon$-NE)* is a strategy profile π in which, for each player i, π_i is an ϵ-BR to π_{-i}.

A *normal-form game* is a single-step extensive-form game. An extensive-form game induces a normal-form game in which the legal actions for player i are its deterministic strategies $\times_{s_i \in \mathcal{I}_i} \mathcal{A}_i(s_i)$. These deterministic strategies are called *pure strategies* of the normal-form game. A *mixed strategy* is a distribution over a player's pure strategies.

Related Work

Double Oracle (DO) and Policy Space Response Oracles (PSRO)

Double Oracle (McMahan, Gordon, and Blum 2003) is an algorithm for finding a NE in normal-form games. The algorithm works by keeping a population of strategies Π^t at time t. Each iteration a NE $\pi^{*,t}$ is computed for the game restricted to strategies in Π^t. Then, a best response to this NE for each player $\mathbb{BR}_i(\pi_{-i}^{*,t})$ is computed and added to the population $\Pi_i^{t+1} = \Pi_i^t \cup \{\mathbb{BR}_i(\pi_{-i}^{*,t})\}$ for $i \in \{1, 2\}$. The DO algorithm is described in Algorithm **??**. Although in the worst case DO must expand all pure strategies, in many games DO empirically terminates early and outperforms existing methods. An interesting open problem is characterizing games where DO will outperform existing methods.

Algorithm 1: Double Oracle

Result: Nash Equilibrium
Input: initial population Π^0
while *Not terminated* **do**
    Solve game restricted to policies in Π^t to get
    meta-distribution π^r
    for $i \in \{1, 2\}$ **do**
        Find a best response $\mathbb{BR}_i(\pi_{-i}^r)$
        $\Pi_i^{t+1} = \Pi_i^t \cup \mathbb{BR}_i(\pi_{-i}^r)$
    end
    if *No novel best response exists for both players*
    then
        Return π^r
    end
end

Policy-Space Response Oracles (PSRO) (Lanctot et al. 2017) approximates the Double Oracle algorithm. The restricted-game NE is computed on the empirical game matrix U^Π, generated by having each policy in the population Π play each opponent policy and tracking average utility in

a $\Pi_1 \times \Pi_2$ payoff matrix (Wellman 2006). In each iteration, an approximate best response to the current restricted NE over the policies is computed via any RL algorithm.

Minimum-Regret Constrained Profile

The concept of finding a low-exploitability distribution in a restricted game was also explored in Jordan, Schvartzman, and Wellman (2010) and Wang, Ma, and Wellman (2021), where they define the *minimum regret constrained profile* as the distribution over a restricted population that achieves the lowest exploitability. In this paper, we use the term *least-exploitable restricted distribution* instead of minimum regret constrained profile. We demonstrate that RM-BR (Johanson et al. 2012) can be used to calculate the least-exploitable restricted distribution, and we incorporate this method into a DO algorithm that converges to an approximate Nash equilibrium through our algorithm RM-BR DO. We also present APSRO, which extends RM-BR DO to large games through RL.

Range of Skill (ROS)

This paper presents ADO, a modification to the *range of skill (ROS)* algorithm introduced in (Zinkevich, Bowling, and Burch 2007) and further explored in (Hansen, Miltersen, and Sørensen 2008). ROS is a variant of the DO algorithm that produces a series of restricted games by iteratively adding new strategies. As in ADO, ROS defines two seperate restricted games where one player is restricted to play strategies in their population while the other player is unrestricted. Also as in ADO, the restricted distribution is part of a Nash equilibrium strategy profile for the restricted game and is therefore the least-exploitable restricted distribution. However, ADO and ROS differ in the strategy that they add to the unrestricted player's population. ADO adds a best response to a restricted Nash equilibrium, while ROS adds a strategy that is part of a restricted NE.

This difference proves crucial when scaling up to large games. In large games, solving the restricted game where one player is unrestricted is infeasible and as a result methods based on ROS cannot scale to large games. Alternatively, since ADO only adds best responses, it naturally scales to large games via APSRO where the best responses are learned through RL. Additionally, while ROS, like ADO, decreases exploitability monotonically and performs well in practice, the only known convergence guarantees for ROS are asymptotic with a convergence rate that is exponential in the size of the game (Hansen, Miltersen, and Sørensen 2008). Conversely, ADO is guaranteed to converge in a number of iterations at most the number of pure strategies in a game. Finally, ROS is only guaranteed to reach an ϵ-Nash, not an exact NE, whereas ADO reaches an exact NE upon termination.

Anytime Double Oracle (ADO)

The double oracle (DO) algorithm is guaranteed to converge because in the worst case it must expand all pure strategies, at which point it terminates at a Nash equilibrium (NE). Unfortunately, before convergence, there is no guarantee on the

exploitability of the restricted-game NE. In fact, DO can increase exploitability from one iteration to the next.

To see this, consider the game in Figure 1. If both players start with a population consisting only of the first strategy (top row and left column), then the best response for each player is the second strategy, giving that player value 1 , for a total exploitability of 2. However, in the next iteration DO will include both the first and second strategies in the population for both players, and the restricted-game NE will give probability 1 to the second strategy. This restricted NE has exploitability of 4. We also provide a generalization to this bad case in Section and show empirically that DO does indeed arbitrarily increase exploitability before terminating in this class of games.

In this paper, we introduce Anytime Double Oracle (ADO), which is guaranteed to not increase exploitability from one iteration to the next. Like DO, ADO maintains a population Π^t at every timestep t. Also like DO, in every iteration a Nash equilibrium is computed on a restricted game and a best response to this restricted NE is computed and added to both populations. However, unlike DO, a different restricted game is created for each player. The restricted game G^i for player i is created by restricting that player to only play strategies included in their population Π_i, while the opponent can play any strategy in the full game. The game value of G^i for player i is

$$\max_{\pi_i \in \Pi_i} \min_{\pi_{-i}} v_i(\pi_i, \pi_{-i}). \tag{1}$$

The restricted game G^i for player i is then solved for both players to get a restricted NE (π_1^i, π_2^i) for both players. We refer to the restricted player's mixed strategy as the restricted NE $\pi_i^r = \pi_i^i$. The restricted NE for player i is the least exploitable mixed strategy supported by player i's population. Note that in large games this restricted game will be prohibitively large to solve, and will require APSRO, introduced later in this paper.

Next, a best response $\beta_i = \mathbb{BR}(\pi_{-i}^r)$ is computed for each player i against the restricted-NE mixed strategy of the restricted opponent, and is added to the player's population. If there are multiple best responses, a novel best response $\beta_i \notin \Pi_i$ is chosen that is not currently in that player's population. ADO is described in Algorithm **??**.

ADO is guaranteed to terminate because there are finitely many pure strategies in the original game. When ADO terminates, the restricted NE is a Nash equilibrium in the original game (Proposition 2). Unlike DO, the exploitability of the restricted NE does not increases from one iteration to the next (Proposition 1).

Proposition 1. *The exploitability of ADO is monotonically non-increasing.*

Proof. Let π^t be the restricted NE in ADO at iteration t. Then for player i, since $\Pi_i^t \subseteq \Pi_i^{t+1}$

$$\begin{aligned}
v_i(\pi_i^t, \mathbb{BR}_{-i}(\pi_i^t)) &= \max_{\pi_i \in \Pi_i^t} \min_{\pi_{-i}} v_i(\pi_i, \pi_{-i}) \\
&\leq \max_{\pi_i \in \Pi_i^{t+1}} \min_{\pi_{-i}} v_i(\pi_i, \pi_{-i}) \tag{2} \\
&= v_i(\pi_i^{t+1}, \mathbb{BR}_{-i}(\pi_i^{t+1})).
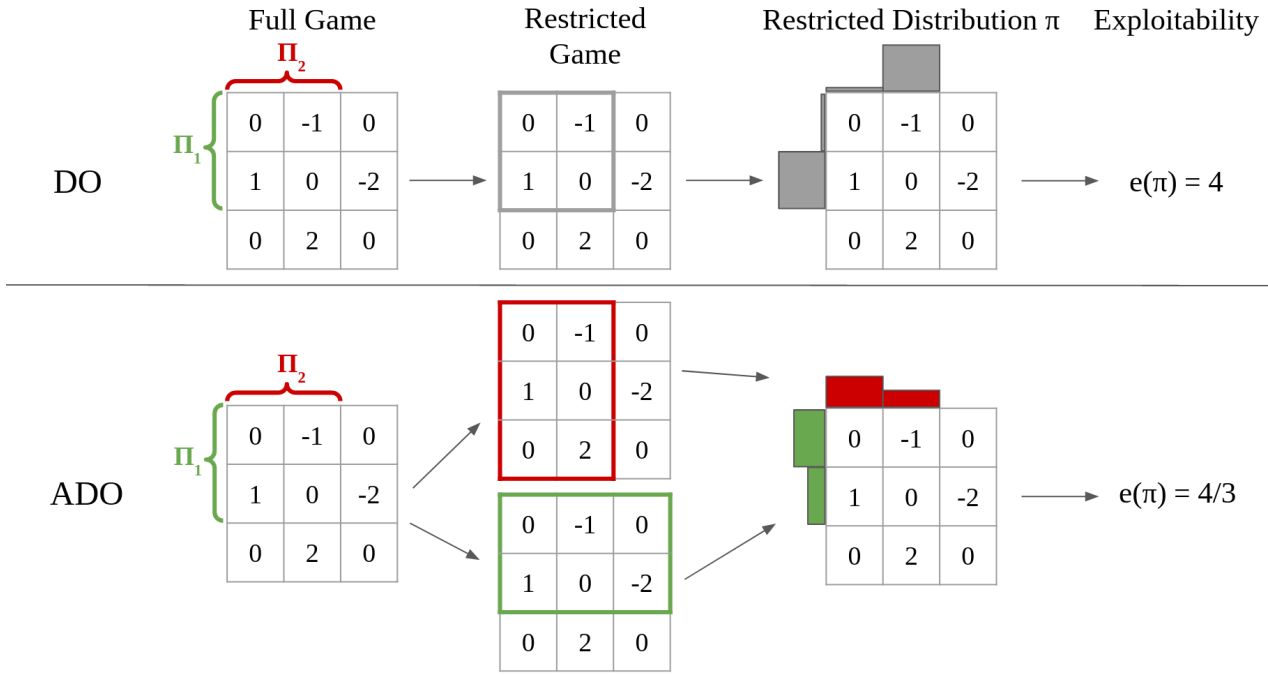\end{aligned}$$

Figure 1: In DO, a single restricted game is created and solved. Since this restricted game does not consider strategies outside of the population, it can lead to exploitable restricted distributions, as shown in the top figure. Conversely, ADO creates two restricted games where the opponent is unrestricted. These restricted games are then solved which results in the least-exploitable restricted distributions for both players.

Algorithm 2: Anytime Double Oracle (ADO)

Result: Nash Equilibrium
Input: initial population Π^0
while *Not terminated* **do**
    for $i \in \{1, 2\}$ **do**
        Solve restricted game defined in equation 1 to
          get meta-distribution π_i^r
        Find a novel best response $\mathbb{BR}_i(\pi_{-i}^r)$
        $\Pi_i^{t+1} = \Pi_i^t \cup \mathbb{BR}_i(\pi_{-i}^r)$
    end
    if *No novel best response exists for both players*
    then
        Return π^r
    end
end

Since each player's value is monotonically non-decreasing, the value of the best response is non-increasing, and the exploitability of π^t is also non-increasing:

$$e(\pi^{t+1}) := -\sum_i v_i(\pi_i^{t+1}, \mathbb{BR}_{-i}(\pi_i^{t+1}))$$
$$\leq -\sum_i v_i(\pi_i^t, \mathbb{BR}_{-i}(\pi_i^t)) = e(\pi^t). \quad (3)$$

□

To illustrate this property of ADO, consider the algorithm dynamics on the DO bad case given in Figure 1. Similar to DO, ADO adds the second strategy to the population in the first iteration. Now, however, instead of taking the second strategy with probability 1 as DO does, ADO solves the restricted game where one player is restricted to the first two strategies and the other is unrestricted and can play all three strategies. The Nash equilibrium of this game for the restricted player is to play the first strategy with probability $\frac{2}{3}$ and to play the second strategy with probability $\frac{1}{3}$. This strategy results in a total exploitability of $\frac{4}{3}$, compared with the DO exploitability of 4 and the initial ADO exploitability of 2. In addition to this property of never increasing exploitability, ADO is guaranteed to converge to a Nash equilibrium, as shown below.

Proposition 2. *When ADO terminates, the restricted NE of both players is a Nash equilibrium in the full game.*
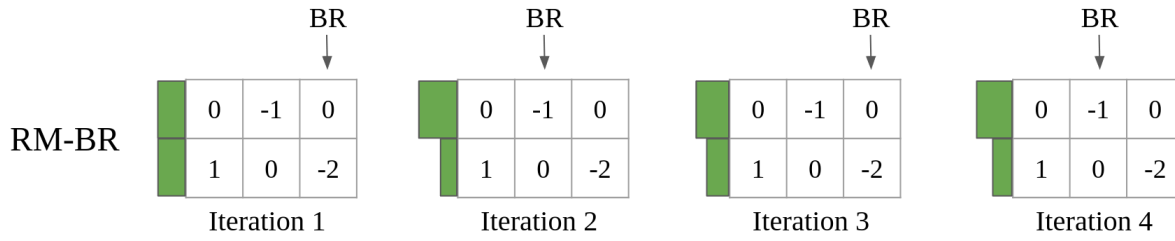
Figure 2: RM-BR converges to the Nash equilibrium of the restricted game by updating the restricted distribution via a no-regret algorithm against a best response at every iteration. Here the best response to the current restricted distribution is indicated by the arrow and the restricted distribution is in green. RM-BR allows us to produce a low-exploitability restricted distribution without solving the full restricted game. This basic idea is the core of APSRO as well, where we use the best response that is being trained as a proxy for the true best response and update the restricted distribution the same way.

Proof. Let (π_1^r, π_2') and (π_1', π_2^r) be the Nash equilibria in the restricted games G^1 and G^2 for player 1 and 2, respectively, upon termination. If π_1' or π_2' have support outside the population, ADO would not terminate, because there would exist another novel best response. So the support of both π_1' and π_2' must be inside the population, which makes them feasible for their respective player's restricted game. Then

$$
\begin{aligned}
v_1(\pi_1^r, \pi_2^r) &\leq v_1(\pi_1', \pi_2^r) \\
&\leq v_1(\pi_1', \pi_2') \\
&\leq v_1(\pi_1^r, \pi_2') \\
&\leq v_1(\pi_1^r, \pi_2^r).
\end{aligned}
\tag{4}
$$

The four inequalities follow, in order, because: (a) player 1 doesn't want to deviate from π_1' to π_1^r in G^2; (b) player 2 doesn't want to deviate from π_2^r to π_2' in G^2; (c) player 1 doesn't want to deviate from π_1^r to π_1' in G^1; and (d) player 2 doesn't want to deviate from π_2' to π_2^r in G^1.

Therefore, $v_1(\pi_1^r, \pi_2^r) = v_1(\pi_1', \pi_2^r)$ which implies that player 1 has no incentive to deviate from π_1^r to π_1' or any other strategy against π_2^r. A symmetric argument holds for player 2, implying that π^r is a Nash equilibrium in the full game. □

Regret-Minimizing against a BR Double Oracle (RM-BR DO)

In this section we propose a specific form of ADO called RM-BR DO, which finds least exploitable restricted strategies π_i^r for the restricted player of each restricted game. RM-BR DO uses a regret minimization vs. a best response (RM-BR) method that is similar to CFR-BR (Johanson et al. 2012): the restricted player uses regret minimization (RM) to find its mixed strategy, alternating with the unrestricted opponent finding best responses (BR). Interestingly, despite not computing the full restricted NE in each restricted game, RM-BR DO is guaranteed to not increase exploitability from one iteration to the next by more than the approximation errors of the RM and BR oracles.

For a given game (later, this will be each restricted game), RM-BR (Johanson et al. 2012) works as follows. First, a regret-minimizing (RM) algorithm, such as regret matching (Hart and Mas-Colell 2000), is initialized. Then, in every

iteration, a best response to the current RM strategy is calculated. Next, the RM strategy is updated with respect to this best response. In this work, we use the Exp3 update rule (Auer et al. 1995) to update the RM strategy, but any regret-minimizing algorithm would work. RM-BR is described in Algorithm 3.

Algorithm 3: RM-BR

Result: Least-Exploitable Meta-strategy of Π_i
Input: initial population Π_i
Initialize π arbitrarily (e.g. uniform random) **for** $t = 1, ..., n$ **do**
   for $i \in \{1, 2\}$ **do**
      Find $\mathbb{BR}_{-i}(\pi_i^r)$
      Sample a strategy from π_i^r
      Observe reward against $\mathbb{BR}_{-i}(\pi_i^r)$
      Update π_i^r via Exp3
   end
end

Proposition 3. *RM-BR with a regret minimizing algorithm that has regret R_t at iteration t will output a distribution π^n such that $e(\pi^n) \leq \frac{R_n}{n}$.*

Proof. The proof follows the same argument as used in Theorem 3 of Johanson et al. (2012). □

In RM-BR DO, we use RM-BR to compute the restricted NE π^r in each iteration of ADO. Formally, RM-BR DO keeps a population of pure strategies Π^t at time t. In each iteration, π^r is initialized to a uniform distribution over pure strategies in both populations, and in an inner loop updated against a series of best responses, as in RM-BR. The final best response to the restricted NE is added as a strategy to the population for both players. The algorithm terminates when the difference between the game value of each player's restricted game is less than ϵ, indicating that π^r is an ϵ-NE in the original game (Hansen, Miltersen, and Sørensen 2008).

Proposition 4. *Assume RM-BR DO runs sufficiently many RM-BR inner-loop updates in each iteration such that the exploitability in each restricted game is at most ϵ. Then the ex-*
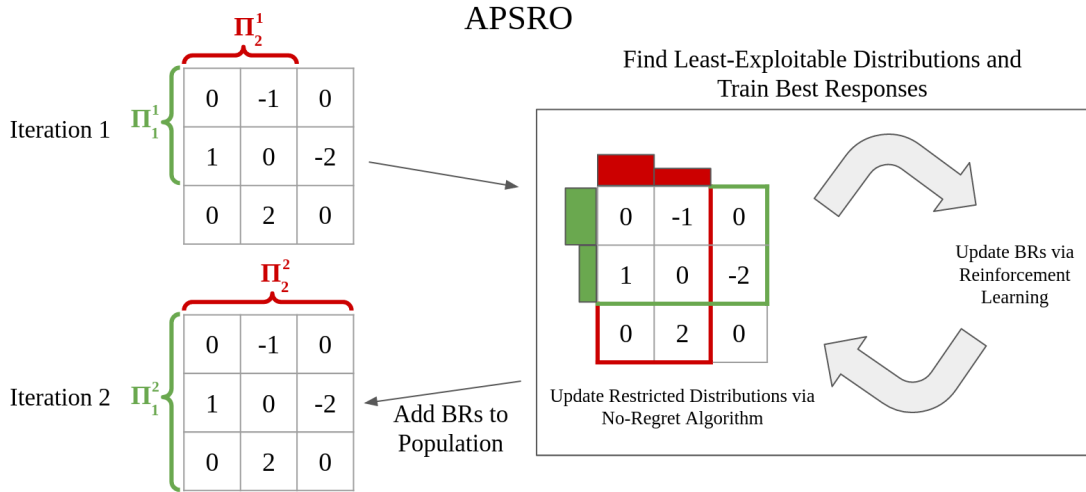
Figure 3: In APSRO, the following happens in one iteration: (1) Two restricted games are created where one player is unrestricted, one for each player. (2) For both players, a BR is trained against the restricted distribution while the restricted distribution is updated via a no-regret algorithm against this BR. (3) This BR is then added to the population.

Algorithm 4: RM-BR DO

Result: Approximate Nash Equilibrium
Input: initial population Π^0
while *Not terminated* **do**
  Get meta-distribution π^r via RM-BR
  Find $\mathbb{BR}_i(\pi^r_{-i})$ for $i \in \{1, 2\}$
  $\Pi_i^{t+1} = \Pi_i^t \cup \mathbb{BR}_i(\pi^r_{-i})$ for $i \in \{1, 2\}$
end

---

ploitability of RM-BR DO will never increase by more than 2ϵ from one iteration to the next.

Proof. Let π^t be the restricted NE of RM-BR DO at iteration t. Then since $\Pi_i^t \subseteq \Pi_i^{t+1}$

$$
\begin{aligned}
v_i(\pi_i^t, \mathbb{BR}_{-i}(\pi_i^t)) &\leq \max_{\pi_i \in \Pi_i^t} \min_{\pi_{-i}} v_i(\pi_i, \pi_{-i}) \\
&\leq \max_{\pi_i \in \Pi_i^{t+1}} \min_{\pi_{-i}} v_i(\pi_i, \pi_{-i}) \qquad (5) \\
&\leq v_i(\pi_i^{t+1}, \mathbb{BR}_{-i}(\pi_i^{t+1})) + \epsilon.
\end{aligned}
$$

$\square$

Anytime PSRO (APSRO)

Although the exploitability of RM-BR DO will never increase much, it needs to compute many different best responses in each inner loop iteration. This can quickly become infeasible, especially in large games. In this section we introduce a scalable version of ADO, called Anytime PSRO (APSRO).

Instead of recomputing a best response in every inner-loop iteration, APSRO maintains one approximate best response RL policy and updates it for a small number of steps

---

Algorithm 5: Anytime PSRO (APSRO)

Result: Meta-Distribution over Population
Input: initial population Π^0
while *Not terminated* **do**
  for $t = 1, ..., n$ **do**
    for $i \in \{1, 2\}$ **do**
      for $j = 1, ..., m$ **do**
        Update $\hat{\mathbb{BR}}_{-i}(\pi_i^r)$ toward a best
          response $\mathbb{BR}_{-i}(\pi_i^r)$
      end
      Sample a strategy from π_i^r
      Observe reward against $\hat{\mathbb{BR}}_{-i}(\pi_i^r)$
      Update π_i^r via Exp3
    end
  end
  $\Pi_i^{t+1} = \Pi_i^t \cup \hat{\mathbb{BR}}_i(\pi^r_{-i})$ for $i \in \{1, 2\}$
end

(a) Random Normal Form Games (b) Leduc with Oracle Best Responses (c) Leduc with Q-Learning Best Responses
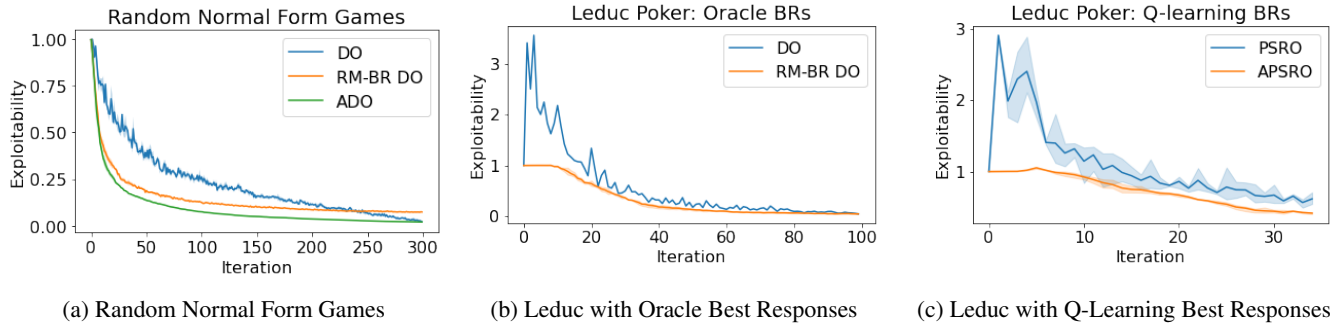
Figure 4: Tabular Methods

in each inner-loop iteration. To speed up the algorithm, AP-SRO can update the approximate best response in fewer iterations than are necessary to get a full best response. This hyperparameter for the number of best-response updates per every Exp3 update trades off between theoretical guarantees and speed.

The updates to the best response can be made through a variety of algorithms. In this paper we show experiments with updates via tabular Q-learning as well as deep reinforcement learning.

Experiments

We report results on normal form games, Leduc poker, Goofspiel, and a continuous-action game introduced in McAleer et al. (2021). Details of these games are included in the appendix.

Experiments with Tabular Methods

In this section we report ADO and RM-BR DO results with oracle best responses and tabular Q learning best responses. In all ADO experiments we solve the restricted games using a linear program.

DO Bad Case To demonstrate how DO can increase exploitability every iteration besides the last, we present a generalization of the game presented in Figure 1. Consider a game where all values are 0, except if the row index r is one more than the column index c, in which case the value is $\sum_{i=0}^{r} 2^i + 2i$, or if the column index c is one more than the row index r, in which case the value is $\sum_{i=0}^{c} -2^i + 2i$. r0.23
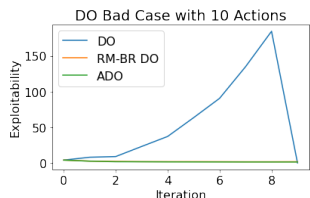
We plot the performance of DO and ADO in this game with 9 actions in Figure and show that DO increases exploitability in every iteration except the last, while ADO does not increase exploitability.

Random Normal-Form Games To create random normal-form games, we sample values from Uniform(0,1). Figure 4a shows results on random normal-form games with 500 actions. We see that ADO greatly outperforms DO and tends to not increase exploitability.

Leduc Poker We test RM-BR DO on Leduc poker in OpenSpiel (Lanctot et al. 2019) and compare it with DO. As shown in Figure 4b, RM-BR DO outperforms DO and does not increase exploitability, but the performance of the two algorithms converges at the end. Although RM-BR DO requires many inner-loop best-response iterations, it is guaranteed to not increase exploitability much from one iteration to the next, and APSRO approximates this process.

Leduc Poker with Tabular Q Learning Best Responses In this section we have a tabular Q learning agent as the best response. Similar to the ADO results, in Figure 4c we see that APSRO is able to perform better than PSRO and does not increase exploitability. However, as in PSRO, the ending exploitability is still much higher than the exploitability that a method like CFR (Zinkevich et al. 2008; Brown and Sandholm 2019; Farina et al. 2019) would achieve in Leduc.

Experiments with Deep Reinforcement Learning

In this section we use deep reinforcement learning for the best response for both APSRO and PSRO, and we present results on Leduc poker, Goofspiel, and a continuous-action hill climbing game. Training details are described in the appendix. To calculate approximate exploitability in Goofspiel and the continuous-action hill climbing game, we train reinforcement learning best responses from scratch against checkpoints.

Leduc Poker As shown in Figure 5a, APSRO with deep RL best responses performs competitively with PSRO but has a much smoother exploitability curve. PSRO with deep RL best responses exhibits a similar fluctuations in exploitability as previously demonstrated with PSRO with tabular Q learning best responses. However, APSRO starts out at a higher exploitability than with tabular Q learning. We conjecture that this is likely due to different initialization between tabular and neural best-response methods. We use DDQN (Van Hasselt, Guez, and Silver 2016) for the deep RL best responses.
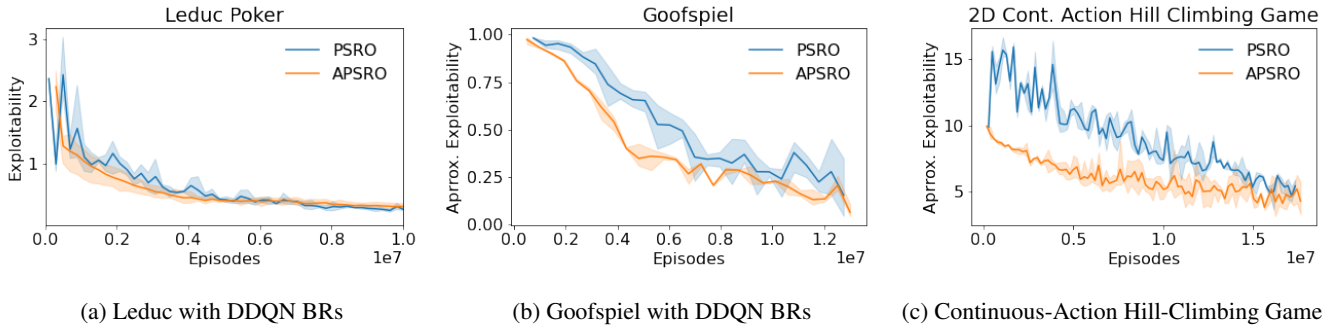
(a) Leduc with DDQN BRs   (b) Goofspiel with DDQN BRs   (c) Continuous-Action Hill-Climbing Game

Figure 5: Deep RL Methods

Goofspiel We experiment on the standard Goofspiel game in Openspiel (Lanctot et al. 2019) with 13 cards and simultaneous moves. As shown in figure 5b, APSRO with deep RL best responses outperforms PSRO and has a slightly smoother exploitability curve. Goofspiel is a much larger game than Leduc poker, suggesting that APSRO can scale to large games. We use DDQN (Van Hasselt, Guez, and Silver 2016) for the deep RL best response.

Continuous-Action Hill-Climbing Game The 2-dimensional continuous-action hill-climbing game was introduced in (McAleer et al. 2021) as a simple continuous-action game where PSRO performs relatively poorly. As shown in Figure 5c, PSRO indeed increases exploitability many times and does not reach the same exploitability as a random initialization until roughly 5,000,000 episodes. Conversely, APSRO rarely increases exploitability and stays below PSRO throughout the entire experiment. We use PPO (Schulman et al. 2017) for the deep RL best response.

Discussion

We introduced ADO, a modification of DO that does not increase exploitability much from one iteration to the next. We also introduced RM-BR DO, a modification of ADO that finds the least exploitable restricted distribution via a no-regret procedure. Finally, we introduced APSRO, a procedure that approximates ADO but can handle reinforcement learning best responses. As shown in our experiments, ADO and RM-BR PSRO outperform DO on normal form games and APSRO outperforms PSRO on Leduc poker. Existing approaches to improving PSRO could potentially be combined with our method (McAleer et al. 2021; Balduzzi et al. 2019; Smith, Anthony, and Wellman 2021; Feng et al. 2021; McAleer et al. 2020; Wright, Wang, and Wellman 2019).

Regarding worst-case convergence time, it is known that some games could have long sequences of subgames, even when the unrestricted subgames are fully solved (Hansen, Miltersen, and Sørensen 2008). This issue is a potential problem for both ADO and DO. How one could reduce the size of these worst-case sequences is still an open question, and an interesting direction for future work.

One downside of ADO compared to DO is that it requires best responses to be novel best responses, and can get stuck at high exploitability if best responses are not chosen to be

novel. This could pose a problem in APSRO, where finding novel best responses could be difficult. One possible method to encourage novelty would be to include a diversity reward for the BR.

References

Auer, P.; Cesa-Bianchi, N.; Freund, Y.; and Schapire, R. E. 1995. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, 322–331. IEEE.

Balduzzi, D.; Garnelo, M.; Bachrach, Y.; Czarnecki, W.; Perolat, J.; Jaderberg, M.; and Graepel, T. 2019. Open-ended learning in symmetric zero-sum games. *International Conference on Machine Learning (ICML)* .

Brown, N.; and Sandholm, T. 2019. Solving imperfect-information games via discounted regret minimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 1829–1836.

Farina, G.; Kroer, C.; Brown, N.; and Sandholm, T. 2019. Stable-predictive optimistic counterfactual regret minimization. In *International conference on machine learning*, 1853–1862. PMLR.

Feng, X.; Slumbers, O.; Yang, Y.; Wan, Z.; Liu, B.; McAleer, S.; Wen, Y.; and Wang, J. 2021. Discovering Multi-Agent Auto-Curricula in Two-Player Zero-Sum Games. *Advances in Neural Information Processing Systems (NeurIPS)* .

Hansen, E. A.; Bernstein, D. S.; and Zilberstein, S. 2004. Dynamic programming for partially observable stochastic games. *Conference on Artificial Intelligence (AAAI)* .

Hansen, T. D.; Miltersen, P. B.; and Sørensen, T. B. 2008. On Range of Skill. *Conference on Artificial Intelligence (AAAI)* .

Hart, S.; and Mas-Colell, A. 2000. A simple adaptive procedure leading to correlated equilibrium. *Econometrica* 68(5): 1127–1150.

Johanson, M.; Bard, N.; Burch, N.; and Bowling, M. 2012. Finding Optimal Abstract Strategies in Extensive Form Games. In *Conference on Artificial Intelligence (AAAI)*.

Jordan, P. R.; Schvartzman, L. J.; and Wellman, M. P. 2010. Strategy exploration in empirical games. In *9th Interna-*

tional Conference on Autonomous Agents and Multiagent Systems (AAMAS). Citeseer.

Lanctot, M.; Lockhart, E.; Lespiau, J.-B.; Zambaldi, V.; Upadhyay, S.; Pérolat, J.; Srinivasan, S.; Timbers, F.; Tuyls, K.; Omidshafiei, S.; et al. 2019. OpenSpiel: A framework for reinforcement learning in games. *arXiv preprint arXiv:1908.09453* .

Lanctot, M.; Zambaldi, V.; Gruslys, A.; Lazaridou, A.; Tuyls, K.; Pérolat, J.; Silver, D.; and Graepel, T. 2017. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

McAleer, S.; Lanier, J.; Baldi, P.; and Fox, R. 2021. XDO: A double oracle algorithm for extensive-form games. *Advances in Neural Information Processing Systems (NeurIPS)* .

McAleer, S.; Lanier, J.; Fox, R.; and Baldi, P. 2020. Pipeline PSRO: A Scalable Approach for Finding Approximate Nash Equilibria in Large Games. In *Advances in Neural Information Processing Systems*.

McMahan, H. B.; Gordon, G. J.; and Blum, A. 2003. Planning in the presence of cost functions controlled by an adversary. *Proceedings of the 20th International Conference on Machine Learning (ICML)* .

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* .

Smith, M. O.; Anthony, T.; and Wellman, M. P. 2021. Iterative Empirical Game Solving via Single Policy Best Response. *International Conference on Learning Representations (ICLR)* .

Van Hasselt, H.; Guez, A.; and Silver, D. 2016. Deep reinforcement learning with double q-learning. In *AAAI conference on artificial intelligence*, volume 30.

Vinyals, O.; Babuschkin, I.; Czarnecki, W. M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D. H.; Powell, R.; Ewalds, T.; Georgiev, P.; et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575(7782): 350–354.

Wang, Y.; Ma, Q.; and Wellman, M. P. 2021. Evaluating Strategy Exploration in Empirical Game-Theoretic Analysis. *Conference on Artificial Intelligence (AAAI)* .

Wellman, M. P. 2006. Methods for empirical game-theoretic analysis. *AAAI conference on artificial intelligence* .

Wright, M.; Wang, Y.; and Wellman, M. P. 2019. Iterated Deep Reinforcement Learning in Games: History-Aware Training for Improved Stability. In *EC*, 617–636.

Zinkevich, M.; Bowling, M.; and Burch, N. 2007. A New Algorithm for Generating Equilibria in Massive Zero-Sum Games. In *Twenty-Second Conference on Artificial Intelligence (AAAI)*.

Zinkevich, M.; Johanson, M.; Bowling, M.; and Piccione, C. 2008. Regret minimization in games with incomplete information. In *Advances in Neural Information Processing Systems (NeurIPS)*.