



**HAL**  
open science

# Global min-max Computation for $\alpha$ -Hölder Games

Aurélien Delage, Olivier Buffet, Jilles Dibangoye

► **To cite this version:**

Aurélien Delage, Olivier Buffet, Jilles Dibangoye. Global min-max Computation for  $\alpha$ -Hölder Games. ICTAI 2023 - 35th IEEE International Conference on Tools with Artificial Intelligence, Nov 2023, Atlanta (Georgia), United States. pp.518-525. hal-04382880

**HAL Id: hal-04382880**

**<https://inria.hal.science/hal-04382880>**

Submitted on 9 Jan 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Global min - max Computation for $\alpha$ -Hölder Games

Aurélien Delage  
CITI  
Insa Lyon  
Villeurbanne, France  
aurelien.delage@insa-lyon.fr

Olivier Buffet  
Inria - CNRS  
Université de Lorraine  
Villers-lès-Nancy, France  
olivier.buffet@inria.fr

Jilles S. Dibangoye  
Bernoulli Institute  
University of Groningen  
Groningen, The Netherlands  
j.s.dibangoye@rug.nl

**Abstract**—min-max optimization problems recently arose in various settings. From Generative Adversarial Networks (GANs) to aerodynamic optimization through Game Theory, the assumptions on the objective function vary. Motivated by the applications to deep learning and especially GANs, most recent works assume differentiability to design local search algorithms such as Gradient Descent Ascent (GDA). In contrast, this work will only require  $\alpha$ -Hölder properties to tackle general game-theoretic problems with poor continuity assumptions. Focusing on the example of problems in which max and min optimization variables live in simplices, we provide a simple algorithm, based on Deterministic Optimistic Optimization (DOO), relying on an outer min-optimization using the solutions of an inner max-optimization. The algorithm is shown to converge in finite time to an  $\epsilon$ -global optimum. Experimental validations are given and the time complexity of our algorithm is studied.

**Index Terms**—zero-sum games, min-max, game theory, non-linear optimization, optimization over simplices

## I. INTRODUCTION

There is a growing interest in min-max problems from various communities. Generative Adversarial Networks [1], [2], fair statistical inference [3], [4], robust decision-making [5], and general resource allocation [6] witness its importance for machine learning. While it has been known for a long time that formulating real-life problems as games is relevant to economics [7], physical phenomena [8] are also interestingly linked to such an optimization problem. Basically, any situation in which two entities are trying to optimize different criteria by interacting with a common system might be seen through a game-theoretic perspective. The players' actions having heterogeneous influences on the system, min-max optimization is a problem whose hypotheses and, thus, difficulty substantially vary depending on the context.

As emphasized in the literature [9], applications of game theory to Deep Learning require modeling more and more complex interactions, making the usual convex-concavity or even differentiability assumptions unfortunately too restrictive. This paper thus provides a first answer to tackle such problems, only requiring mild  $\alpha$ -Hölder properties.

Moving on to a formal problem definition, for any given function  $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$  (where  $\mathcal{X} \subset \mathbb{R}^n$  and  $\mathcal{Y} \subset \mathbb{R}^p$ ),

the general min - max problem is to compute at least one pair  $(\mathbf{x}^*, \mathbf{y}^*)$  such that

$$\mathbf{x}^* \in \arg \min_x \max_y f(\mathbf{x}, \mathbf{y}), \quad (1)$$

$$\mathbf{y}^* \in \arg \max_y f(\mathbf{x}^*, \mathbf{y}). \quad (2)$$

The simplest case, introduced by von Neumann [10], assumes that (i)  $\mathcal{X}$  and  $\mathcal{Y}$  are respectively the unit simplices of  $\mathbb{R}^n$  and  $\mathbb{R}^p$ , and (ii)  $f$  is bilinear. Then, min-max equals max-min, the problem corresponds to finding a Nash Equilibrium of a 2-player 0-sum Normal-Form Game, and is solved by a simple linear program in polynomial time [11]. Even though (i) and (ii) can be respectively replaced by (iii)  $\mathcal{X}$  and  $\mathcal{Y}$  convex and (iv)  $f$  convex-concave with min-max still being equal to max-min [12], such equality and simplicity of resolution does not hold in the general case.

Relaxing assumptions (iii) and (iv) on  $f$  leads to various concerns. If  $f$  is not convex or not concave, Nash Equilibria do not necessarily exist [9]. Even if they do, the computational cost of finding one might be very high, as deciding whether a Nash Equilibrium exists when the game is not convex-concave is NP-Hard in general [13].

Assuming only differentiability (but not that a Nash equilibrium necessarily exists), the most common approach to solve the min-max problem is to alternate between gradient ascents and descents to respectively comply with max and min's will. The resulting algorithm is known as *Gradient Descent Ascent* (GDA). However, this can fail [14], even for some simple zero-sum bilinear games [15].

For this reason, under continuous differentiability assumption, a First-order Nash Equilibrium (FNE) solution concept was defined through the first-order Taylor approximation of the function, and the existence of at least one FNE is guaranteed for twice-differentiable functions [16]. Recent work introduced modifications to the GDA algorithm [1], [14], [16] to provide algorithms asymptotically converging towards a FNE in a number of gradient evaluations going from  $\mathcal{O}(\epsilon^{-2})$  to  $\mathcal{O}(\epsilon^{-6})$  [17] through  $\mathcal{O}(\epsilon^{-4})$  [18], [19], depending on the assumptions made on  $f$ . As a matter of fact, computing  $\epsilon$ -FNEs has been shown to be in ExpTime with respect to either  $1/\epsilon$ , the smoothness of the game, or its dimensionality [13].

The intensive study of gradient-based approaches is partly due to its utility for deep learning. Thus, min-max problems without, at least, differentiability assumptions were not studied even though they naturally appear [9, Figure 1,b] [20], [21]. Since we are interested in games with poor continuity properties, we will focus on the min-max solution concept, which is well-defined and exists.

Encouragingly, the min-max problem always admits a solution (by  $f$ 's continuity) and the optima of a "local" min-max version of the problem [18], *i.e.* players can only deviate within a ball of radius  $\delta$  from their strategy, are strongly linked to the stable limit point of GDA [18]. Besides, the min-max computation makes sense as it corresponds to finding *security values* for player min by searching for the most rewarding strategy she can announce to player max, and the resulting problem is called a *zero-sum Stackelberg game*.

For now, relaxing assumptions leads to instabilities and divergent behaviors of gradient-based approaches. To tackle this problem, we build on a global optimizing algorithm for functions with weak continuity assumptions, namely Munos' Deterministic Optimistic Optimization (DOO) [22], to  $\epsilon$ -optimally solve the min-max problem in finite time for functions with only mild  $\alpha$ -Hölder continuity properties. This allows to deal with a larger class of games, starting with the ones including entropy measures in the payoff function [9, Figure 1,b] [20], [21].

After providing some technical background regarding game theory and DOO in Section II, Section III introduces two adaptations of DOO in order to provide a finite-time convergent algorithm whenever the variables live in simplices. Section IV then derives an algorithm to  $\epsilon$ -optimally solve the min-max problem, and the resulting algorithm is experimentally studied in Section V.

*Problem statement and contributions:* Let us now clearly state the problem of interest in this paper and our contributions.

*Problem 1:* Given an  $\alpha$ -Hölder function  $f$ , the problem to solve is computing  $\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} f(x, y)$ . We require (i) an  $\epsilon$ -optimum guarantee and (ii) finite-time convergence.

Our contributions related to this problem are to: (i) provide a finite-time converging algorithm towards an  $\epsilon$ -optimum based on the single variable optimizing algorithm DOO (see Section II); (ii) show how it can be implemented in practice if  $\mathcal{X}$  and  $\mathcal{Y}$  are simplices (Section III-B); (iii) study the more complex case of dependent feasible set games (*i.e.* computing  $\min_{x \in \mathcal{X}} \max_{y \in F(x) \subset \mathcal{Y}} f(x, y)$ ) (Section IV-B).

## II. BACKGROUND

First, we start by providing some necessary background about game theory and insights about the relevance of the  $\alpha$ -Hölder condition in games.

### A. Games and solution concepts

*Definition 1 (Two player zero-sum Game):* A two player zero-sum game is defined by a tuple  $\langle \mathcal{A}^1, \mathcal{A}^2, f \rangle$  where, for each  $i \in \{1, 2\}$ , there exists  $p_i \in \mathbb{N}$  such that  $\mathcal{A}^i$  is a compact subset of  $\mathbb{R}^{p_i}$ , and  $f : \mathcal{A}^1 \times \mathcal{A}^2 \rightarrow \mathbb{R}$  is continuous but not

necessarily differentiable. The game will be called a *normal-form game* if (i)  $\mathcal{A}^1$  and  $\mathcal{A}^2$  are respectively the unit simplex  $S_{p_1}$  of  $\mathbb{R}^{p_1}$  and the unit simplex  $S_{p_2}$  of  $\mathbb{R}^{p_2}$  and (ii)  $f$  is a matrix belonging to  $\mathcal{M}_{p_1, p_2}$ .

A simple, yet interesting, example consists in a zero-sum normal-form game  $\langle S_2, S_2, M \rangle$  in which player 1 (resp. 2) also wants to minimize (resp. maximize) the entropy of her mixed strategy, while not degrading too much their expected payoff.

*Example 1 ([9]):* Consider the classical matching pennies game given by the payoff matrix  $M$

$$M = \begin{pmatrix} 1/2 & -1/2 \\ -1/2 & 1/2 \end{pmatrix}, \quad (3)$$

and the bilinear payoff function  $\mathbf{x}^\top \cdot M \cdot \mathbf{y}$ . The payoff function of the modified matching pennies game in which player 1 is rewarded for a high entropy strategy but 2 is penalized for a high entropy strategy is formally given by  $f(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \cdot M \cdot \mathbf{y} + x_1 \log(x_1) + x_2 \log(x_2) + y_1 \log(y_1) + y_2 \log(y_2)$ . In this zero-sum game, no Nash equilibrium exists, and the payoff function is neither convex nor concave. Still, the payoff function is  $\alpha$ -Hölder for any  $\alpha \in ]0, 1[$ .

*Definition 2 ( $\alpha$ -Hölder condition):* For any norm  $\|\cdot\|$ , a function  $f : \mathcal{X} \rightarrow \mathbb{R}$  satisfies the  $\alpha$ -Hölder condition with respect to  $\|\cdot\|$  if and only if there exists  $\alpha \in ]0, 1[$  and  $C \in \mathbb{R}^+$  such that

$$\forall (\mathbf{x}, \tilde{\mathbf{x}}) \in \mathcal{X}^2, |f(\mathbf{x}) - f(\tilde{\mathbf{x}})| \leq C \|\mathbf{x} - \tilde{\mathbf{x}}\|^\alpha. \quad (4)$$

As mentioned in the introduction, Nash Equilibrium points (*i.e.* joint strategies which no player has incentive in unilaterally deviating from) do not necessarily exist, in which case  $\max\text{-min} > \min\text{-max}$ , even in a local sense [18]. Players thus can not agree on a joint strategy, and might want to search for the individual strategy that has the greatest value against any strategy of their opponent, *i.e.* (for player 1) compute  $\arg \min_{\mathbf{x}} \max_{\mathbf{y}} f(\mathbf{x}, \mathbf{y})$ .

Since this paper aims to tackle games with poor continuity properties of the payoff function, we will only assume that  $f$  is  $\alpha$ -Hölder<sup>1</sup>.

### B. Deterministic Optimistic Optimization (DOO)

Let  $f : \mathcal{X} \rightarrow \mathbb{R}$  be a  $\lambda$ -Lipschitz function with respect to any semi-metric  $l$  (*i.e.*  $\forall (\mathbf{x}, \mathbf{x}') \in \mathcal{X}^2, |f(\mathbf{x}) - f(\mathbf{x}')| \leq \lambda l(\mathbf{x}, \mathbf{x}')$ ) defined over  $\mathcal{X} \subset \mathbb{R}^n$ . Given any semi-metric  $l$ , a cell (compact subset)  $S$  and point  $\tilde{\mathbf{x}}$  in  $S$ ,  $f$ 's Lipschitz continuity with respect to  $l$  allows optimistically bounding, *i.e.* lower bounding, its value within  $S$  by  $f(\tilde{\mathbf{x}}) - \lambda \text{Diam}(S)$ , where  $\text{Diam}(S) = \max_{\mathbf{x}, \mathbf{y} \in S} l(\mathbf{x}, \mathbf{y})$ . The smaller the diameter of the cell, the closer the optimistic bound is to the values of  $f$  in it.

If  $\mathcal{X}$  is bounded, DOO  $\epsilon$ -optimally solves  $\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$  for a given error  $\epsilon > 0$  by creating a non-uniform covering of the domain with a finite number of cells where the variations

<sup>1</sup>For the sake of simplicity, we only expose the results for  $\alpha$ -Hölder functions, even though Munos [22] presents the result for Lipschitz functions with respect to any semi metric  $l$  (not just  $\|\cdot\|^\alpha$ ).

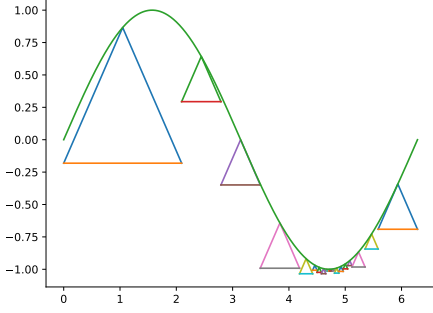


Fig. 1: Representation of DOO’s execution to minimize  $x \mapsto \sin(x)$  on  $[0, 2\pi]$ . The interval is covered with cells (*i.e.* here, intervals) of different sizes where the function  $\sin$  is lower-bounded by “Lipschitz cones”, themselves lower-bounded by a constant. The cones, along with their constant lower bounds, form the triangular shapes.

of the function are controlled. Let us assume that  $\mathcal{X}$  is such that (i) there is an analytic way to create a first covering with cells, and (ii) each cell can again be covered with children cells. The algorithm (given in Algorithm 1) starts with the first covering of  $\mathcal{X}$  (line 2), and computes the optimistic bound in every cell (line 3). Then, each iteration consists in selecting the most promising cell according to the optimistic bound (line 6), covering it (line 7), and computing the optimistic bound of each new cell (line 8). The final returned value is the encountered point with the lowest value. Figure 1 gives an illustration of an execution of DOO.

---

#### Algorithm 1: DOO

---

```

1 Fct DOO ( $[\mathcal{X} \rightarrow \mathbb{R}; x \mapsto f(x)]$ ,  $n$ )
   input :  $f : \mathbb{R} \rightarrow \mathbb{R}$   $\alpha$ -Hölder func. with constant  $C$ 
2   Initialize  $\mathcal{I}$  and  $(\mathcal{S}_i)_{i \in \mathcal{I}}$  s.t.  $\mathcal{X} \subseteq \cup_{i \in \mathcal{I}} \mathcal{S}_i$ 
3    $\forall i \in \mathcal{I}$ , compute  $f(\mathbf{x}_i) - C\rho_i^\alpha$ 
   /*  $\rho_i$  : diameter of  $\mathcal{S}_i$ ;
   /*  $\mathbf{x}_i$  : representative element of  $\mathcal{S}_i$ . */
4    $time \leftarrow 0$ 
5   while  $time \leq n$  do
6      $i^* \leftarrow \arg \min_{i \in \mathcal{I}} f(\mathbf{x}_i) - C\rho_i^\alpha$ 
7     Recover  $\mathcal{S}_{i^*}$  by  $\cup_{j \in \mathcal{I}^*} \mathcal{S}_j$  ( $\supseteq \mathcal{S}_{i^*}$ )
   /*  $\forall j \in \mathcal{I}^*, \mathbf{x}_j \leftarrow Repr(\mathcal{S}_j)$ 
   /*  $Repr(\mathcal{S}_j)$  : point  $\in \mathcal{S}_j \cap \mathcal{X}$ . */
8      $\mathcal{I} \leftarrow [\mathcal{I} \setminus \{i^*\}] \cup \mathcal{I}^*$ 
9      $time \leftarrow time + 1$ 
10  return  $\langle \arg \& \min_{x_i} f(x_i) \rangle$ 

```

---

If the cells are “well-formed” (following Assumption 1 below) and their diameter decreases w.r.t their depth in the tree (following Assumption 2), the approximation error shrinks with the radius around the most promising area as the algorithm iterates. Thus, the difference  $r(n)$  between the smallest

value of  $f$  and DOO’s returned value after performing  $n$  coverings of cells with children cells can be bounded [22].

*Assumption 1 (Valid cell (Assumption 4 in [22])):* There exists  $\nu \in \mathbb{R}^{+,*}$  such that, for any cell  $\mathcal{S} \subset \mathcal{X}$ , there exists  $\mathbf{x} \in \mathcal{S}$  such that  $\mathcal{B}_l(\mathbf{x}, \nu \text{Diam}(\mathcal{S})) \subset \mathcal{S}$ , where  $\mathcal{B}_l(\mathbf{x}, \rho) = \{y \in \mathcal{X} | l(\mathbf{x}, y) \leq \rho\}$  denotes the ball of radius  $\rho$  centered in  $\mathbf{x}$  for the semi-metric  $l$ .

*Assumption 2:* There exists a decreasing sequence  $\delta(h)$  such that, for any depth  $h \geq 0$ , for any cell  $\mathcal{S}_{h,i}$  of depth  $h$  in the tree,  $\text{Diam}(\mathcal{S}_{h,i}) \leq \delta(h)$ .

*Remark 1:* Fortunately, for any given norm  $\|\cdot\|$ , (i) an  $\alpha$ -Hölder function with constant  $C$  is exactly a  $C$ -Lipschitz function with respect to the semi-metric  $\|\cdot\|^\alpha$ , and (ii) balls  $\mathcal{B}_{\|\cdot\|^\alpha}(\mathbf{x}, \rho)$  are valid cells. We are now able to adapt Munos’ work to tackle min-max problems for  $\alpha$ -Hölder functions, using simple balls  $\mathcal{B}_{\|\cdot\|^\alpha}(\cdot, \cdot)$ .

The next two sections respectively show how DOO can be (i) applied to the special case of optimization over a simplex and (ii) adapted to solve a min-max problem.

### III. FINITE-TIME CONVERGENT DOO FOR SIMPLEX SPACES

In this section, we come back to vanilla minimization (rather than min-max optimization) and provide two modifications to the original DOO algorithm in order to match usual requirements of game theory. First, we require our algorithm to converge in finite time to an  $\epsilon$ -optimum instead of bounding the regret for a given time budget. This leads to introducing a pessimistic bound which will also be useful later as a pruning criterion. Secondly, we show that DOO can be used when the search space is a simplex by applying iterative simplex discretization methods.

#### A. Modifying the stopping criterion

The initial analysis of DOO’s complexity bounds its error with respect to a given budget  $n$ . We change viewpoint to show that, for any  $\epsilon > 0$ , one can derive a stopping criterion that is reached in finite time. To do so, we introduce upper and lower bounds of the optimal value, which will monotonically shrink as the algorithm iterates. The algorithm returns whenever the difference between the maximum value of the upper-bound and the maximum value of the lower-bound is smaller than  $\epsilon$ .

*Definition 3 (Upper and lower bounds):* Let  $f$  be an  $\alpha$ -Hölder function and  $C$  be an  $\alpha$ -Hölder constant of  $f$ . Let  $(\mathcal{B}_{\|\cdot\|^\alpha}(\mathbf{x}_i, \rho_i))_i$  be a set of cells partitioning  $f$ ’s domain. Then,  $\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$  is upper- and lower-bounded respectively by  $\min_{\mathbf{x}_i} f(\mathbf{x}_i)$  and  $\min_{\mathbf{x}_i} f(\mathbf{x}_i) - C\rho_i^\alpha$ .

A straightforward adaptation of Munos’ regret-bounding proof leads to the following convergence theorem.

*Theorem 1 (adapted from [22]):* Let  $\epsilon > 0$  be a given error. Let  $(\mathcal{B}_{\|\cdot\|^\alpha}(\mathbf{x}_i, \rho_i))_i$  be the set of cells covering  $f$ ’s domain, updated as the algorithm iterates. The quantity  $|\min_i [f(\mathbf{x}_i) - C\rho_i^\alpha] - \min_i [f(\mathbf{x}_i)]|$  decreases throughout the optimization process and is smaller than  $\epsilon$  after a finite number of iterations.

*Remark 2:* One could derive a precise complexity bound, but at the cost of introducing function-dependent topological constants whose computation is, in general, even harder than the optimization problem. Instead, Theorem 3 (Section IV-A) will provide an upper bound of the complexity, based on the worst-case scenario of optimizing a constant function.

Note that (i)  $\min_i f(\mathbf{x}_i) - C\rho_i^\alpha$  and  $\min_i f(\mathbf{x}_i)$  have no reason to be attained in the same cell; and (ii) Theorem 1 leads to simply modifying line 5 of Algorithm 1 to stop whenever the gap between bounds is less than  $\epsilon$ . The parameter  $n$  in Algorithm 1 is thus replaced by  $\epsilon$ .

### B. DOO for simplex spaces

a) *Subdivision process:* We now detail how to subdivide the  $n$ -dimensional unit simplex  $S_n(1)$  using smaller and smaller hypercubes as the process iterates.

Even though simplex decomposition techniques are not new [23], [24], we introduce a new simple subdivision process that is well suited to the use of DOO. The process starts with the  $n$ -dimensional hypercube (i.e. the closed ball  $\mathcal{B}_{\|\cdot\|_\infty}(\mathbf{0}, 1) = \{x \in \mathbb{R}^n \mid \|x - \mathbf{0}\|_\infty \leq 1\}$  centered in  $\mathbf{0}$  with a radius 1). Note that hypercubes are cells that cover  $[0, 1]^n$  whereas their intersection with the unit simplex are cells covering the  $n - 1$  dimensional simplex. The following theorem shows how to determine whether a hypercube intersects the simplex or not, and how to compute a *referent* point, on which the function will be evaluated. All the created hypercubes can again be covered with  $2^n$  ones, and so on, which creates an iterative covering of the unit simplex with smaller and smaller hypercubes.

*Theorem 2 (Intersection between the  $n$ -dimensional unit simplex and an  $n$ -dimensional hypercube):* Let  $n \in \mathbb{N}^* \setminus \{1\}$ . Let  $H$  be an  $n$ -dimensional cube (i.e. a closed ball for  $\|\cdot\|_\infty$ ) of radius  $\eta$  whose center is called  $m$ . Then,  $H$  and  $S_n(1)$  intersect (i.e.,  $(H \cap S_n(1)) \neq \emptyset$ ) if and only if  $\exists (x_i, x_j) \in H^2$  such that  $\sum_{k=1}^n x_i^k \leq 1$  and  $\sum_{k=1}^n x_j^k \geq 1$ , and an intersection point (called referent point) can be computed.

*Proof 1:*

Let us consider the diagonal from the lowest point ( $\mathbf{x}_{inf} = (m^1 - \eta, \dots, m^n - \eta)$ ) to the highest point ( $\mathbf{x}_{sup} = (m^1 + \eta, \dots, m^n + \eta)$ ) of  $H$ , and apply the Intermediate Value Theorem. There is no intersection point if  $\sum_k \mathbf{x}_{inf}^k > 1$  or  $\sum_k \mathbf{x}_{sup}^k < 1$ . Else, the intersection point is  $x = \mathbf{x}_{inf} + t(\mathbf{x}_{sup} - \mathbf{x}_{inf})$ , where  $t = \frac{1 - \sum_{k=1}^n \mathbf{x}_{inf}^k}{2n\eta}$ . Indeed,

$$\begin{aligned} \sum_{k=1}^n x^k = 1 &\iff \sum_{k=1}^n [\mathbf{x}_{inf}^k + t(\mathbf{x}_{sup}^k - \mathbf{x}_{inf}^k)] = 1 \\ \iff t &= \frac{1 - \sum_{k=1}^n \mathbf{x}_{inf}^k}{\sum_k \mathbf{x}_{sup}^k - \mathbf{x}_{inf}^k} \iff t = \frac{1 - \sum_{k=1}^n \mathbf{x}_{inf}^k}{n \cdot 2\eta}. \end{aligned}$$

Figure 3 illustrates the iterative subdivision of the 2-dimensional unit simplex by 2-dimensional hypercubes (i.e., squares). Let us point out that the subdivision operation is here concentrated around the optimum:  $(0, 1)$ . Squares with an ‘x’ do not intersect with the simplex and are pruned,

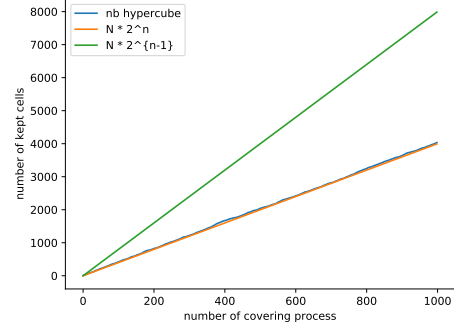


Fig. 2: Number of hypercubes kept when optimizing over the  $n$ -dimensional simplex (here,  $n = 3$ ) as a function of DOO’s number of iterations ( $N$ ).

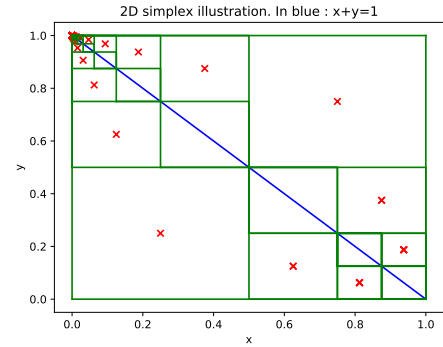


Fig. 3: Illustration of the subdividing process

while the other ones do and are therefore kept. The unit-simplex being of dimensionality  $n - 1$  and the hypercubes of dimensionality  $n$ , as illustrated by Figure 2, we conjecture that  $\lim_{n \rightarrow \infty} \#H(N) = N \cdot 2^{n-1}$ , where  $\#H(N)$  denotes the number of hypercubes kept after  $N$  iterations.

## IV. MIN-MAX $\alpha$ -HÖLDER OPTIMIZATION

In this section, we come back to the min-max problem to provide a global optimization algorithm that converges in finite time to an  $\epsilon$ -optimal solution.

To do so, one can first notice the  $\alpha$ -Hölder continuity of the outer function  $\mathbf{x} \mapsto \max_{\mathbf{y}} f(\mathbf{x}, \mathbf{y})$ .

*Lemma 1:* Let  $f$  be a  $\alpha$ -Hölder function and let  $C$  be one of its  $\alpha$ -Hölder constants. The function  $\mathbf{x} \mapsto \max_{\mathbf{y}} f(\mathbf{x}, \mathbf{y})$  is  $\alpha$ -Hölder and  $C$  is one of its  $\alpha$ -Hölder constants.

*Proof 2:* Let

$$g(\mathbf{x}) \stackrel{\text{def}}{=} \max_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}).$$

Then, for any  $\mathbf{x}$  and  $\mathbf{x}'$ , assuming wlog that  $g(\mathbf{x}) > g(\mathbf{x}')$ ,

$$\begin{aligned} g(\mathbf{x}) - g(\mathbf{x}') &= \max_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}) - \max_{\mathbf{y}} f(\mathbf{x}', \mathbf{y}) \\ &\leq \max_{\mathbf{y}} (f(\mathbf{x}', \mathbf{y}) + C\|\mathbf{x}, \mathbf{y}\| - \max_{\mathbf{y}} f(\mathbf{x}', \mathbf{y})) \\ &= \max_{\mathbf{y}} (f(\mathbf{x}', \mathbf{y})) + C\|\mathbf{x} - \mathbf{x}'\| - \max_{\mathbf{y}} f(\mathbf{x}', \mathbf{y}) \\ &= C\|\mathbf{x} - \mathbf{x}'\|. \end{aligned}$$

A similar result holds if  $g(\mathbf{x}') > g(\mathbf{x})$ , concluding the proof.

With this property, we solve our min-max optimization problem by using two nested DOO processes, *i.e.*,

- an outer  $\epsilon_1$ -optimal DOO minimizing the function  $\mathbf{x} \mapsto \min_{\mathbf{y}} f(\mathbf{x}, \mathbf{y})$ , using the solution of
- an inner  $\epsilon_2$ -optimal DOO maximizing the function  $\mathbf{y} \mapsto f(\mathbf{x}, \mathbf{y})$  for fixed  $\mathbf{x}$

(see Algorithm 2). Munos' proof can be adapted to this case to show that the final error is  $\epsilon = \epsilon_1 + \epsilon_2$ .

---

### Algorithm 2: BiDOO

---

```

1 Fct
  BiDOO ( $[\mathcal{A}^1 \times \mathcal{A}^2 \rightarrow \mathbb{R}; x, y \mapsto f(x, y)], \epsilon_1, \epsilon_2, C$ )
2   input :  $f : \mathcal{A}^1 \times \mathcal{A}^2 \rightarrow \mathbb{R}$  an  $\alpha$ -Hölder function
    $\langle x_{\min}, v_{\min} \rangle \leftarrow$  DOO (
      $[\mathcal{A}^1 \rightarrow \mathbb{R}; x \mapsto -\text{getMin}(\left($ 
       DOO ( $[\mathcal{A}^2 \rightarrow \mathbb{R}; y \mapsto -f(x, y)], \epsilon_2, C$ )
      $\left. \right)], \epsilon_1, C$ )
   /* getMin( $\cdot, \cdot$ ) here returns its second argument,
     i.e., the minimum of the inner DOO
     computation. */
3 return  $\langle x_{\min}, v_{\min} \rangle$ 

```

---

*Remark 3 (Pruning the inner process):* Using a pessimistic bound for the outer process is not only useful to provide a finite-time convergent algorithm, but also helps to prune some irrelevant parts of the search space. The current pessimistic bound of the outer DOO is passed to the inner DOO when called, so that it stops whenever the value of  $x$  is necessarily higher than the pessimistic bound of the inner process (process not shown in Alg. 2).

#### A. Complexity analysis

Let us now upper bound the number of iterations it takes for BiDOO to converge by analyzing it in the worst case, *i.e.*, optimizing a constant function. Let us assume that BiDOO is called to solve  $f : S_n \times S_p \mapsto \{a\}$  with  $a \in \mathbb{R}$ , but is given an (overestimating)  $\alpha$ -Hölder constant  $C > 0$ . In this case, BiDOO iteratively covers up the simplex with an increasingly thin uniform grid, and the exact number of iterations it takes before reaching a stopping criterion  $\epsilon$  can be found analytically.

*Theorem 3 (Complexity upper-bound):* For any zero-sum game  $\langle S_n, S_p, f \rangle$ , BiDOO returns an  $\epsilon$ -optimum  $(x^*, y^*)$  in less than

$$\left[ \sum_{i=0}^{\log_2(C^\alpha/\epsilon)} (2^n)^i \right] \times \left[ \sum_{i=0}^{\log_2(C^\alpha/\epsilon)} (2^p)^i \right]$$

subdivision processes, where  $C$  is any  $\alpha$ -Hölder constant of  $f$  given to BiDOO.

*Proof 3 (sketch):* BiDOO will subdivide the search space. For a given  $\epsilon$ , and a given overestimating  $\alpha$ -Hölder constant  $C$ , the minimum diameter of the cells to ensure an  $\epsilon$ -convergence is  $\epsilon/C^\alpha$ . With hypercubes, the radius of the cells is divided by 2 at each subdividing process, so that a depth of  $\log_2(\epsilon/C^\alpha)$  in DOO's tree is required. Now, each evaluation of a node in BiDOO's outer tree has, at worst, the same complexity as a complete  $2^p$ -ary tree of depth  $\log_2(\epsilon/C^\alpha)$ . The outer tree after convergence being, at worst, a  $2^n$ -ary tree of depth  $\log_2(\epsilon/C^\alpha)$ , the complexity result holds. The worst case is obtained when  $f$  is constant.

#### B. Games with dependent feasible set

In this section, we will look at games with dependent feasible sets, formally given by the computation of

$$\min_{x \in \mathcal{A}^1} \max_{y \in F(x) \subset \mathcal{A}^2} f(x, y) \quad (5)$$

for some  $F$ . Such games received a growing interest recently, especially in the form of Fisher market problems [25], formulated as a game with dependent feasible set in [26]. Assuming that  $f$  is  $\alpha$ -Hölder in the whole space  $\mathcal{A}^1 \times \mathcal{A}^2$ , the following provides a sufficient hypothesis on  $F$  to ensure BiDOO's convergence on such a game.

*Theorem 4:* Let  $\mathcal{B}(\mathcal{A}^2)$  denote the Borelian algebra of  $\mathcal{A}^2$ . If,  $\forall x \in \mathcal{A}^1$ ,  $F(x)$  is a bounded, closed and convex (*i.e.* compact convex) element  $\mathcal{B}(\mathcal{A}^2)$  and  $F$  is  $\lambda$ -Lipschitz continuous for the Hausdorff distance, then BiDOO converges towards an  $\epsilon$ -optimum in finite time.

*Proof 4:*

Let us start by defining the Hausdorff distance for compact sets of  $\mathcal{B}(\mathcal{A}^2)$ :

$$H : \mathcal{B}(\mathcal{A}^2)^2 \rightarrow \mathbb{R}^+$$

$$(A, B) \mapsto \max\{\sup_{a \in A} \inf_{b \in B} \|a - b\|_\infty, \sup_{b \in B} \inf_{a \in A} \|a - b\|_\infty\}.$$

Let  $f : \mathcal{A}^1 \times \mathcal{A}^2 \rightarrow \mathbb{R}$  be  $\alpha$ -Hölder, and  $C$  be a Hölder constant of  $f$ . Assume that  $F : \mathcal{A}^1 \rightarrow \mathcal{B}(\mathcal{A}^2)$  is  $\lambda$ -Lipschitz-continuous for the Hausdorff distance. Let  $x \in \mathcal{A}^1$ . Let  $\epsilon > 0$ . We aim at bounding the variations of  $x \mapsto \max_{y \in F(x)} f(x, y)$  within a ball of radius  $\epsilon$  around  $x$ , formally given by  $\sup_{\tilde{x} \in B_{\|\cdot\|_\infty}(x, \epsilon)} |\max_{y \in F(x)} f(x, y) - \max_{y \in F(\tilde{x})} f(\tilde{x}, y)|$ .

A first observation is that,  $\forall \tilde{x} \in B_{\|\cdot\|_\infty}(x, \epsilon)$ ,

$$\forall y \in F(\tilde{x}), f(\tilde{x}, y) \leq f(x, y) + C\|x - \tilde{x}\|_\infty^\alpha,$$

since  $f$  is  $\alpha$ -Hölder in the whole space  $\mathcal{A}^1 \times \mathcal{A}^2$ , so that

$$\max_{y \in F(\tilde{x})} f(\tilde{x}, y) \leq \max_{y \in F(\tilde{x})} [f(x, y)] + C\|x - \tilde{x}\|_\infty^\alpha.$$

Then, for any  $\tilde{x} \in B_{\|\cdot\|_\infty}(x, \epsilon)$ , let us define  $I$  as

$$I \stackrel{\text{def}}{=} \left| \max_{y \in F(\tilde{x})} f(x, y) - \max_{y \in F(x)} f(x, y) \right|. \quad (6)$$

Let  $y^* \in \arg \max_{y \in F(x)} f(x, y)$  and  $\tilde{y}^* \in \arg \max_{y \in F(\tilde{x})} f(x, y)$  (whose existence is given by  $f$ 's continuity on a compact set).

Assuming wlog that  $f(x, y^*) \geq f(x, \tilde{y}^*)$ , we have

$$|I| = f(x, y^*) - \max_{y \in F(\tilde{x})} f(x, y) \quad (7)$$

$$\leq f(x, y^*) - f(x, \Pi_{F(\tilde{x})}(y^*)), \quad (8)$$

where  $\Pi_{F(\tilde{x})}(y) \in \arg \min_{\beta \in F(\tilde{x})} \|y^* - \beta\|_\infty$ , which exists as a minimum as we are dealing with compact sets of  $\mathbb{R}^n$ . Now,

$$\|y^* - \Pi_{F(\tilde{x})}(y^*)\|_\infty \leq \sup_{y \in F(\tilde{x})} \|y - \Pi_{F(\tilde{x})}(y)\|_\infty \quad (9)$$

$$\leq H(F(x), F(\tilde{x})). \quad (10)$$

Thus,

$$f(x, y^*) - f(x, \Pi_{F(\tilde{x})}(y^*)) \leq CH(F(x), F(\tilde{x}))^\alpha \quad (11)$$

$$\leq C\lambda \|x - \tilde{x}\|_\infty^\alpha \quad (12)$$

and, finally,  $|I| \leq C\lambda \|x - \tilde{x}\|_\infty^\alpha \leq C\lambda \epsilon^\alpha$  so that  $\forall \tilde{x} \in B_{\|\cdot\|_\infty}(x, \epsilon)$ ,

$$\left| \min_{y \in F(x)} f(x, y) - \min_{y \in F(\tilde{x})} f(\tilde{x}, y) \right| \leq C\|x - \tilde{x}\|_\infty^\alpha + C\lambda \|x - \tilde{x}\|_\infty^\alpha$$

$$\leq C\epsilon \left( 1 + \frac{\epsilon^\alpha}{\epsilon} \right).$$

This defines the upper bound of a cell  $B_{\|\cdot\|_\infty}(x, \epsilon)$  so that DOO applies to the computation of the outer function  $g : x \mapsto \max_{y \in F(x)} f(x, y)$ .

## V. EXPERIMENTS

The following aims at studying the behavior of BiDOO (i) relatively to its hyperparameters (Secs. V-A and V-B) and (ii) for games with dependent feasible set (Section V-C). All experiments ran on an Ubuntu machine with i7-10810U 1.10 GHz Intel processor, 16 GB available RAM and the code used is available at <https://github.com/aurelienDelageInsaLyon/minimaxDOO> under MIT license.

When considering normal-form games, the exact min-max value is given by the resolution of a linear program using Cplex<sup>2</sup>, and the  $\alpha$ -Hölder constants of the inner and the outer processes are obtained using the following lemma.

*Lemma 2:*  $f : x \mapsto \max_{\mathbf{y}} \mathbf{x}^\top \cdot M \cdot \mathbf{y}$  is  $\sum_j \max_i |m_{i,j}|$ -Lipschitz and,  $\forall x, \mathbf{y} \mapsto \mathbf{x}^\top \cdot M \cdot \mathbf{y}$  is  $\sum_i \max_j |m_{i,j}|$ -Lipschitz when  $x, \mathbf{y}$  are elements of the unit simplex of  $\mathbb{R}^n$ .

<sup>2</sup>Cplex 12.1 <https://www.ibm.com/fr-fr/analytics/cplex-optimizer>

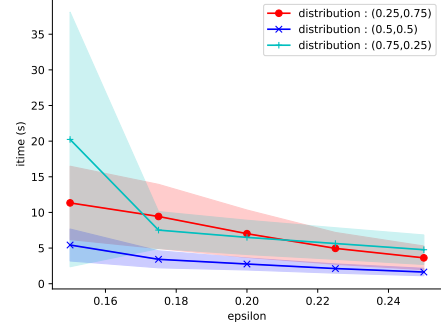


Fig. 4: BiDOO's convergence time (average and standard error) as a function of  $\epsilon$  for 3 different distributions.

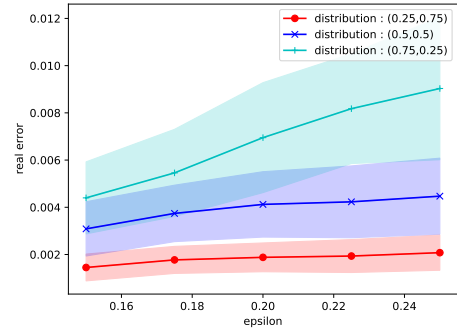


Fig. 5: BiDOO's observed error (average and standard error) as a function of  $\epsilon$  for 3 different distributions.

### A. Choosing the $\epsilon$ -distribution

A question is whether one should choose  $\epsilon_1 = \epsilon_2 = \frac{1}{2}\epsilon$  or another distribution, such as  $\epsilon_1 = \frac{3}{4}\epsilon$  or  $\epsilon_2 = \frac{1}{4}\epsilon$ . To partially study this, Figures 4 and 5 respectively provide BiDOO's computation time and actual error when solving the min-max problem for randomly generated  $3 \times 3$  matrices taken in  $[0, 1]^{3 \times 3}$ . Results are randomized over 30 matrices. We observe that (i) with the 50/50 distribution, BiDOO appears to converge faster, but (ii) with the 25/75 distribution, BiDOO provides a lower error in average. While the error results appear very intuitive, the timing ones are a bit more surprising. One could expect the 25/75 distribution BiDOO to be faster than the 50/50 one as it is less demanding with the inner process, which is called exponentially more often than the outer one. Still, as the number of iterations required to provably reach an error less than a given  $\epsilon = \epsilon_1 + \epsilon_2$  is exponential w.r.t  $\epsilon$ , the 25/75 distribution BiDOO may require way more time for the outer DOO to converge. This might, as a whole, take more time. In all the following experiments, we pick  $\epsilon_1 = \epsilon_2$ .

### B. Validating the approach

We consider 100 randomly generated matrices  $M \in [0, 1]^{3 \times 3}$ . Figures 6 and 7 respectively give the error compared



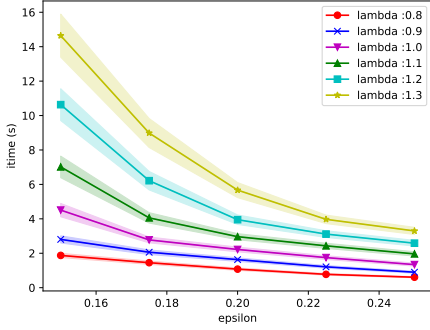


Fig. 6: Computation time (in seconds) for Algorithm 2 as a function of the error  $\epsilon$  imposed, for different Hölder constants. If  $C^1$  and  $C^2$  denote the 1-Hölderian constants of the outer and the inner DOO processes, the curve labelled “lambda : 1.1” corresponds to launching BiDOO with  $C^1 \cdot 1.1$  and  $C^2 \cdot 1.1$ .

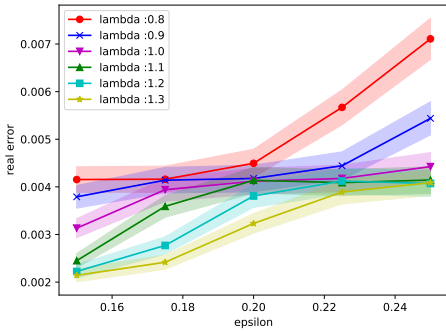


Fig. 7: Error of Algorithm 2 compared to the exact value as a function of the error  $\epsilon$  imposed, for different Hölder constants. If  $C^1$  and  $C^2$  denote the 1-Hölderian constants of the outer and the inner DOO processes, the curve labelled “lambda : 1.1” corresponds to launching BiDOO with  $C^1 \cdot 1.1$  and  $C^2 \cdot 1.1$ .

to the exact value and the computation time as a function of the error imposed to the algorithm.

Several things are to be noted on this figure. Firstly, we observe consistency, *i.e.* BiDOO’s error with respect to the exact value always being lower than the error  $\epsilon$  imposed to BiDOO. Secondly, and as predicted by Section IV-A, the time taken by the algorithm to converge appears exponential with respect to  $\epsilon$ . Interestingly, BiDOO returns a value that is guaranteed to be  $\epsilon$ -close to the optimal, but which is actually around  $\epsilon/10$ -close to the exact value.

Looking back at Theorem 3, we compared a uniform search iteratively building a small enough uniform grid to BiDOO using the same set of 100 randomly-generated  $3 \times 3$  matrices, for  $\epsilon = 0.15$ . It takes the uniform search in average 7.8 s (std. error 0.9 s) to end, whereas BiDOO converges on average in 2.4 s (std. error 0.2 s).

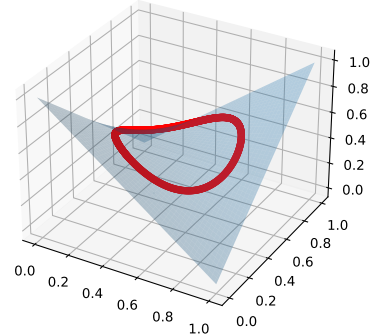


Fig. 8: GDA applied to the problem  $\max_x \min_y x \cdot y^\top$ . Red points correspond to points visited by the GDA algorithm.

### C. Comparison with the state of the art

a) *Bilinear Games and Gradient Descent Ascent*: It is known [27] that “vanilla” GDA (*i.e.* with constant step size) can fail even for the bilinear game  $x^\top \cdot I_n \cdot y$ , where  $x, y \in \mathbb{R}^n$ , and  $I_n$  is the  $n$ -dimensional identity matrix. An illustration of GDA’s behavior is given in Figure 8. As the game is bilinear, it is *a fortiori*  $\alpha$ -Hölder, and BiDOO provably solves the min-max problem.

b) *Games with dependent feasible sets*: The difficulties with running a gradient-based algorithm to solve general min-max problems can be illustrated [14] through the computation of

$$\min_{x \in [-1, 1]} \max_{y \in [-1, 1], 1 - (x+y) \geq 0} x^2 + y + 1. \quad (13)$$

In such a game, GDA converges to  $(0, 1)$ , which is not a solution of the min-max problem, whereas BiDOO does converge as illustrated in Figure 9, provided the subdivision process is modified to satisfy the constraint  $y \in [-1, 1], 1 - (x+y) \geq 0$ . This result is expected as one could show that this example satisfies the assumptions of Section IV-B.

Interestingly, Figure 9 shows a pessimistic bound within  $\epsilon$  of the optimum way earlier than the optimistic bound. This behavior is frequent with approaches guaranteeing  $\epsilon$ -optimality using optimistic and pessimistic bounds; proving optimality (*i.e.*, lowering the optimistic bound  $\epsilon$ -closely to the optimum) is harder than heuristically finding near-optimal points.

## VI. DISCUSSION

The time complexity of the proposed algorithm was shown to be exponential with respect to (i) the dimension of the game and (ii)  $1/\epsilon$ . This theoretical claim is supported by our experiments. This is an expected result as computing a local Nash Equilibrium, when the game has stronger properties than the  $\alpha$ -Hölder one (differentiability and Lipschitz gradient) [9], is already in ExpTime with respect to either the game’s dimension,  $1/\epsilon$  or the game’s smoothness. We introduced the



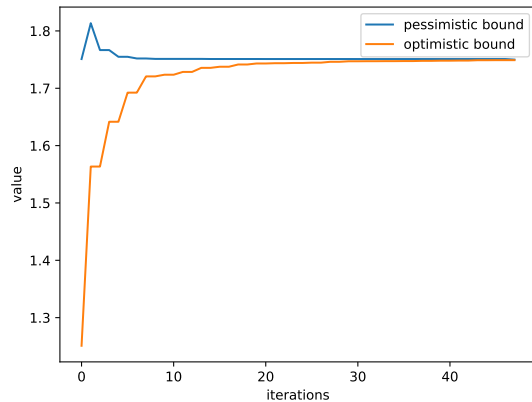


Fig. 9: Upper and lower bounds as a function of the iteration number.

BiDOO algorithm, which is (to the best of our knowledge) the first algorithm to provably converge towards an  $\epsilon$ -min – max of any  $\alpha$ -Hölder game. The algorithm being simple, we believe that it may inspire further work. As an example, if the game’s payoff is convex with respect to player 2, then the inner DOO process can be replaced by a quicker gradient descent. Moreover, our algorithm can be extended to the unknown smoothness case, building on SOO [22] instead of DOO. BiDOO might also pave the way for tackling general-sum games with any number of players with independent divergent payoff functions by extending our algorithm to multi-level optimization problems. Still, this remains an open question as the function giving 1’s value for a strategy  $x$  given that 2 best responds to it according to her payoff function is discontinuous, and the computation of discontinuities is a harder problem than computing a Stackelberg Equilibrium. Showing that there only exists a finite number of discontinuities (which was proven for the one-dimensional case for a dense set of payoff functions [28]) should enable the use of an  $\epsilon$ -greedy version of BiDOO asymptotically converging towards an  $\epsilon$ -Stackelberg equilibrium for any  $\alpha$ -Hölder game.

## VII. CONCLUSION

This paper proposes a method to deal with min-max optimization of  $\alpha$ -Hölder functions, leading to a finite-time convergent algorithm towards an  $\epsilon$ -global optimum. We focused on the optimization over simplices, derived an algorithm and evaluated it. We hope our method can be useful whenever such a problem arises.

## REFERENCES

- [1] M. Sanjabi, J. Ba, M. Razaviyayn, and J. D. Lee, “On the convergence and robustness of training GANs with regularized optimal transport,” *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [2] F. A. Oliehoek, R. Savani, J. Gallego-Posada, E. Van der Pol, E. D. De Jong, and R. Groß, “GANs: Generative adversarial network games,” *arXiv preprint arXiv:1712.00679*, 2017.
- [3] P. Sattigeri, S. C. Hoffman, V. Chenthamarakshan, and K. R. Varshney, “Fairness GAN,” *arXiv preprint arXiv:1805.09910*, 2018.

- [4] M. Jagielski, M. Kearns, J. Mao, A. Oprea, A. Roth, S. Shari-falvajardi, and J. Ullman, “Differentially private fair learning,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 3000–3008.
- [5] Y. Chow, A. Tamar, S. Mannor, and M. Pavone, “Risk-sensitive and robust decision-making: A CVaR optimization approach,” *Advances in neural information processing systems*, vol. 28, 2015.
- [6] J. Du, L. Zhao, J. Feng, and X. Chu, “Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee,” *IEEE Transactions on Communications*, vol. 66, no. 4, pp. 1594–1608, 2017.
- [7] D. Friedman, “On economic applications of evolutionary game theory,” *Journal of Evolutionary Economics*, vol. 8, no. 1, pp. 15–43, 1998.
- [8] N. Brunner and N. Linden, “Connection between Bell nonlocality and Bayesian game theory,” *Nature communications*, vol. 4, no. 1, pp. 1–6, 2013.
- [9] C. Daskalakis, “Non-concave games: A challenge for game theory’s next 100 years,” *Nobel symposium “One Hundred Years of Game Theory: Future Applications and Challenges*, 2021.
- [10] J. von Neumann, “Zur Theorie der Gesellschaftsspiele,” *Mathematische Annalen*, vol. 100, 1928. [Online]. Available: <https://doi.org/10.1007/BF01448847>
- [11] Y. Shoham and K. Leyton-Brown, *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.
- [12] M. Sion, “On general minimax theorems,” *Pacific Journal of mathematics*, vol. 8, no. 1, pp. 171–176, 1958.
- [13] C. Daskalakis, S. Skoulakis, and M. Zampetakis, “The complexity of constrained min-max optimization,” in *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, 2021, pp. 1466–1478.
- [14] D. Goktas and A. Greenwald, “Gradient descent ascent in min-max Stackelberg games,” *arXiv preprint arXiv:2208.09690*, 2022.
- [15] L. Mescheder, A. Geiger, and S. Nowozin, “Which training methods for GANs do actually converge?” 2018.
- [16] M. Nouiehed, M. Sanjabi, T. Huang, J. D. Lee, and M. Razaviyayn, “Solving a class of non-convex min-max games using iterative first order methods,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [17] H. Rafique, M. Liu, Q. Lin, and T. Yang, “Weakly-convex concave min-max optimization: Provable algorithms and applications in machine learning,” 2018.
- [18] C. Jin, P. Netrapalli, and M. I. Jordan, “What is local optimality in nonconvex-nonconcave minimax optimization?” 2019.
- [19] T. Lin, C. Jin, and M. Jordan, “On gradient descent ascent for nonconvex-concave minimax problems,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 6083–6093.
- [20] E. Asarin, J. Cervelle, A. Degorre, C. Dima, F. Horn, and V. Kozyakin, “Entropy games and matrix multiplication games,” *arXiv preprint arXiv:1506.04885*, 2015.
- [21] S. Brandsen, I. J. Geng, and G. Gour, “What is entropy? A perspective from games of chance,” *Physical Review E*, vol. 105, no. 2, p. 024117, 2022.
- [22] R. Munos, “Optimistic optimization of a deterministic function without the knowledge of its smoothness,” *Advances in Neural Information Processing Systems*, vol. 24, 2011.
- [23] H. Edelsbrunner and D. R. Grayson, “Edgewise subdivision of a simplex,” in *Proceedings of the fifteenth annual symposium on Computational geometry*, 1999, pp. 24–30.
- [24] R. Paulavicius and J. Žilinskas, *Simplicial global optimization*. Springer, 2014.
- [25] I. Fisher, “Mathematical investigations in the theory of value and prices, and appreciation and interest,” Ph.D. dissertation, Yale University, 1892.
- [26] D. Goktas and A. Greenwald, “Convex-concave min-max Stackelberg games,” *NeurIPS*, vol. 34, pp. 2991–3003, 2021.
- [27] J. Zhang, P. Xiao, R. Sun, and Z. Luo, “A single-loop smoothed gradient descent-ascent algorithm for nonconvex-concave min-max problems,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 7377–7389, 2020.
- [28] A. Bressan and Y. Jiang, “On the generic structure and stability of stackelberg equilibria,” *Journal of Optimization Theory and Applications*, vol. 183, pp. 840–880, 2019.