OJMO

Open Journal of Mathematical Optimization

Jérôme Bolte, Lilian Glaudin, Edouard Pauwels & Mathieu Serrurier

The backtrack Hölder gradient method with application to min-max and min-min problems

Volume 4 (2023), article no. 8 (17 pages)

https://doi.org/10.5802/ojmo.24

Article submitted on September 2, 2021, revised on April 18, 2023,
accepted on May 6, 2023.

CENTRE MERSENNE

Open Journal of Mathematical Optimization is a member of the
Centre Mersenne for Open Scientific Publishing
http://www.centre-mersenne.org/
e-ISSN: 2777-5860

The backtrack Hölder gradient method with application to min-max and min-min problems

Jérôme Bolte
Toulouse School of Economics, Université Toulouse Capitole, Toulouse, France

Lilian Glaudin
ANITI, University of Toulouse, Toulouse France

Edouard Pauwels
Toulouse School of Economics, Institut Universitaire de France, Toulouse, France.

Mathieu Serrurier
Université Paul-Sabatier, IRIT Toulouse, France

Abstract

We present a new algorithm to solve min-max or min-min problems out of the convex world. We use rigidity assumptions, ubiquitous in learning, making our method – the backtrack Hölder algorithm applicable to many optimization problems. Our approach takes advantage of hidden regularity properties and allows us, in particular, to devise a simple algorithm of ridge type. An original feature of our method is to come with automatic step size adaptation which departs from the usual overly cautious backtracking methods. In a general framework, we provide convergence theoretical guarantees and rates. We apply our findings on simple Generative Adversarial Network (GAN) problems obtaining promising numerical results. It is worthwhile mentioning that a byproduct of our approach is a simple recipe for general Hölderian backtracking optimization.

Digital Object Identifier 10.5802/ojmo.24

Keywords Hölder gradient, backtracking line search, min-max optimization, ridge method, semi-algebraic optimization.

Acknowledgments The authors are grateful to the anonymous referees and M. Friedlander for his precious advices. The authors also acknowledge the support of ANR-3IA Artificial and Natural Intelligence Toulouse Institute. JB and EP thank Air Force Office of Scientific Research, Air Force Material Command, USAF, under grant numbers FA9550-19-1-7026, FA8655-22-1-7012, ANR MaSDOL 19-CE23-0017-0, ANR Regulia, ANR Bonsai. JB acknowledges the support of ANR Chess, grant ANR-17-EURE-0010, Centre Lagrange and TSE-Partnership.

1 Introduction, meta-data commands

The main motivation for our algorithmic developments and analyses relates to applications of adversarial learning, introduced in [19], see also [3], which call for the development of algorithms addressing large scale, smooth problems of the type

$$\min_{x\in\mathbb{R}^d} \max_{y\in\mathcal{Y}} L(x,y), \tag{1}$$

where $\mathcal{Y}$ is a constraint set, and L is a given cost function. This structure happens to be ubiquitous in optimization and game theory, but generally under assumptions that are not those met in learning. In optimization it stems from the Lagrangian approach and duality theory [7, 11, 13], while in game theory it comes from zero-sum 2-players games [25, 36, 37]. Dynamics for addressing (1) have thus naturally two types. They may be built on strategic considerations, so that algorithms correspond to a sequence of actions chosen by antagonistic players, see [25]. In general these methods are not favorable to optimization because the contradictory interests of players induce oscillations and slowness in the identification of optimal strategies. Optimization algorithms seem more interesting for our purposes because they focus on the final result, i.e., finding an optimal choice x, regardless of the adversarial strategy issues. In that respect, there are two possibilities: the variational inequality approach, see e.g. [13, 23, 28] or [18, 22, 27] in learning, which treats minimization and maximization variables on an equal footing. On the other hand, some methods break this symmetry, such as primal or augmented Lagrangian

methods. In these methods, a large number of explicit steps, implicit steps, or global minimization are performed on one variable while the other is updated much more cautiously in an explicit incremental way, see e.g., [7, 34].

Our work is written in this spirit: we assume that the inner-max argument is tractable, that is y solvable, with a good precision, and we construct our algorithm on the following model:

$$\begin{aligned} y_n &= \operatorname{argmax}_{y\in\mathcal{Y}} L(x_n, y), \\ x_{n+1} &= x_n - \gamma_n \nabla_x L(x_n, y_n), \gamma_n > 0,\ n \geq 0. \end{aligned} \tag{2}$$

Note that we use equality sign for the argmax as we will be working under this assumption (see Assumption 4). This assumption ensures that the partial maximum is differentiable in the classical sense and allows to deploy techniques from smooth optimization. Our analysis relies on exact partial maximum evaluation, we leave the analysis of the impact of inexact evaluation as future research. As explained above, the rationale is not new[1], and is akin to many methods in the literature on learning where the global optimization is performed approximately by multiple gradient steps [30, 31] or by clever revision steps, as in the "follow the ridge" method [38].

Backtrack Hölder

What is new then? This brings us to our main abstract contribution, which can be applied beyond min-max structured problems and constitutes a result of independent interest. Surprisingly we can provide theoretical grounds to devise *large steps* and thus obtain aggressive learning-rates with very mild assumptions. Differential regularity of L does not directly translate into regularity of the inner max in (1). However, one can exploit some hidden properties of the value function $g = \max_y L(\,\cdot\,, y)$ under widespread rigidity assumptions. Let us sketch the ideas of our approach. First, under a uniqueness assumption on the maximizer, our method appears to be a gradient method on the value function for "player I" ("the generator" of Generative Adversarial Networks)

$$x_{n+1} = x_n - \gamma_n \nabla g(x_n).$$

Secondly, we use the fact that g has a locally Hölderian gradient[2] whenever L is semialgebraic or analytic-like, a situation which covers most of the problems met in practice. With such observations, our second main contribution is an automatic learning-rate strategy based on a diagonal backtracking method for objectives with Hölderian gradients. The term *diagonal* refers to the idea that the two parameters describing Hölder regularity of the gradient vary together during the search process. The overall procedure consists in applying this idea to the function g which results in an automatic learning-rate strategy for algorithm (2) to solve problem (1). We call this approach "Backtrack Hölder methods for min-max".

Summary of the contributions

- An automatic diagonal step size strategy is devised for gradient descent on objective with qualitative Hölderian gradient and unknown constant and exponent. This is of independent interest as it is not specific to min-max problems.
- Under mild rigidity assumptions the value function g of the inner max in Problem (1) benefits from a qualitative Hölderian gradient property. This results in a new algorithm "Backtrack Hölder for min-max", with automatic step size tuning, which is the main motivation for our developments.
- Our algorithms are shown to find ϵ critical points with at most $O(\epsilon^{-[2+c]})$ iterations, where $c \geq 0$ is related to the Hölderian regularity of the underlying function. Most importantly, we provide general convergence guarantees to points (x^*, y^*) satisfying $y^* = \operatorname{argmax}_y L(x^*, y)$ and $\nabla_x L(x^*, y^*) = 0$. This is done within a fairly general framework, since L is merely assumed semialgebraic while the "best response" of player II is only required to be singled-valued.
- A byproduct of our work is a global convergence result for Hölderian gradient methods (without backtracking) [6, 20, 29, 39].
- Our work is theoretical in essence. It is merely a first step towards convergence analysis of more involved algoriths involving nonsmoothness or stochastic subsampling. We propose, however, numerical experiments on learning problems, including "Sinkhorn GANs", [16, 17], which rely on optimal transport losses regularized through the addition of an entropic term, and Wasserstein GANs [3], which are natural extensions of GANs [19].

[1] It can be traced back to the origin of augmented Lagrangian methods, see e.g., [32]

[2] Recall that $G\colon \mathbb{R}^d \mapsto \mathbb{R}^{d'}$ is locally Hölderian if for all bounded subset $V \subset \mathbb{R}^d$, there exists β and ν positive such that $\|G(x) - G(y)\| \leq \beta \|x - y\|^\nu$, whenever $x, y \in V$.

2 A recipe for Hölderian gradient descent

We generalize some results of [5, 9] to the Hölderian setting.

2.1 Semialgebraic considerations

Our results often use semialgebraic assumptions which are pervasive in optimization and machine learning [12].

Definition 1 (Semialgebraic sets and functions)**.**

1. *A subset S of $\mathbb{R}^m$ is a* real semialgebraic set *if there exist two positive integers r and s, and, for every $k \in \{1,\dots,r\}$, $l \in \{1,\dots,s\}$, two real polynomial functions P_{kl}, $Q_{kl} \colon \mathbb{R}^m \to \mathbb{R}$ such that*

$$S = \bigcup_{k=1}^{r} \bigcap_{l=1}^{s} \{x \in \mathbb{R}^m : P_{kl}(x) < 0, Q_{kl}(x) = 0\}.$$

2. *A function $f : A \subset \mathbb{R}^m \to \mathbb{R}^n$ is semialgebraic if its graph $\left\{(x,\lambda) \in A \times \mathbb{R}^n \;\middle|\; f(x) = \lambda\right\}$ is a semialgebraic subset of $\mathbb{R}^{m+n}$.*

For illustrations of this notion in large-scale optimization and machine learning we refer to [4, 12]. One will also find in this work references, definitions and examples of globally subanalytic sets that are necessary for our proofs to apply to Sinkhorn GANs (see Remark 17). The following result can be found, for example, in [8].

For illustrations of this notion in large-scale optimization and machine learning we refer to [4, 12]. One will also find in this work references, definitions and examples of globally subanalytic sets that are necessary for our proofs to apply to Sinkhorn GANs (see Remark 17). The following result can be found, for example, in [8].

Proposition 2 (Continuity and semialgebraicity implies Hölder continuity)**.** *Let $f \colon \mathbb{R}^d \to \mathbb{R}^{d'}$ be a semialgebraic continuous function. Then f is locally Hölder, i.e., for all compact sets K,*

$$\exists\, \beta \in]0,+\infty[, \exists\, \nu \in]0,1], \forall\, x, y \in K, \quad \|f(x) - f(y)\| \leq \beta \|x - y\|^{\nu}.$$

We recall below the Łojasiewicz inequality [24].

Definition 3 (Łojasiewicz inequality)**.** *A differentiable function $f \colon \mathbb{R}^n \to \mathbb{R}$ has the Łojasiewicz property at $x^* \in \mathbb{R}^n$ if there exist $\eta, C \in]0,+\infty[$ and $\theta \in]0,1[$ such that for all $x \in B(x^*,\eta)$,*

$$C|f(x) - f(x^*)|^{\theta} \leq \|\nabla f(x)\|.$$

In this case the set $B(x^*,\eta)$ is called a Łojasiewicz ball.

2.2 A convergence recipe for Hölder gradient methods

Let us recall and adapt to the Hölder case a few classical results for Lipschitz gradients. We first recall [39, Lemma 1].

Lemma 4 (Hölder Descent Lemma)**.** *Let $U \subset X$ be a nonempty convex set, let $f \colon \mathbb{R}^d \to \mathbb{R}$ be a C^1 function, let $\nu \in]0,1]$, and let $\beta \in]0,+\infty[$. Suppose that*

$$\forall\, (x,y) \in U^2, \quad \|\nabla f(x) - \nabla f(y)\| \leq \beta \|x - y\|^{\nu}.$$

Then

$$\forall\, (x,y) \in U^2, \quad f(y) \leq f(x) + \langle \nabla f(x), x - y \rangle + \frac{\beta}{\nu+1} \|y - x\|^{\nu+1}.$$

Lemma 5 (Controlled descent)**.** *Let $\delta, \theta \in]0,1[$, let $C, \gamma \in]0,+\infty[$ and let $x, y, x^* \in \mathbb{R}^d$. Suppose that the following hold:*

1. *$f(x) \geq f(x^*)$ and $f(y) \geq f(x^*)$,*
2. *$f(y) \leq f(x) - \frac{\delta}{\gamma}\|y - x\|^2$,*
3. *$\|\nabla f(x)\| \leq \frac{1}{\gamma}\|y - x\|$,*
4. *$C(f(x) - f(x^*))^{\theta} \leq \|\nabla f(x)\|$.*

Then

$$\delta C(1-\theta)\|y-x\| \leq (f(x)-f(x^*))^{1-\theta} - (f(y)-f(x^*))^{1-\theta}.$$

Proof. First, if $y = x$, then the inequality holds trivially. Second, if $f(x) = f(x^*)$, then by the first two items, $y = x$ and the inequality holds also. Hence we may suppose that $f(x) - f(x^*) > 0$ and $y \neq x$. We have

$$\frac{C\gamma}{\|y-x\|} \leq \frac{C}{\|\nabla f(x)\|} \leq (f(x)-f(x^*))^{-\theta}.$$

By concavity of $s \mapsto s^{1-\theta}$, we have

$$\begin{aligned}(f(x)-f(x^*))^{1-\theta} - (f(y)-f(x^*))^{1-\theta} &\geq (1-\theta)(f(x)-f(x^*))^{-\theta}(f(x)-f(y))\\ &\geq \frac{C\gamma(1-\theta)}{\|y-x\|}\frac{\delta}{\gamma}\|y-x\|^2\\ &\geq \delta C(1-\theta)\|y-x\|,\end{aligned}$$

which concludes the proof. ◀

We combine a recipe idea from [9] with a trap argument from [5], extending the elements developed in [1].

Theorem 6 (Recipe for convergence and the trapping phenomenon)**.** *Let $f\colon \mathbb{R}^d \to \mathbb{R}$ be a C^1 function and let $\delta \in]0,1[$. Consider $(x_n)_{n\in\mathbb{N}}$ in $\mathbb{R}^d$ and $(\gamma_n)_{n\in\mathbb{N}}$ in $]0,+\infty[$ that satisfies*

1. $(\forall\, n \in \mathbb{N})\ f(x_{n+1}) \leq f(x_n) - \frac{\delta}{\gamma_n}\|x_{n+1}-x_n\|^2$,

2. $(\forall\, n \in \mathbb{N})\ \|\nabla f(x_n)\| \leq \frac{1}{\gamma_n}\|x_{n+1}-x_n\|$.

Then the following results hold:

i. *Assume that there exist $x^* \in \mathbb{R}^d$, $\theta \in]0,1[$, $\eta, C \in]0,+\infty[$, and $N \in \mathbb{N}$ such that*

$$C|f(x)-f(x^*)|^\theta \leq \|\nabla f(x)\|, \forall\, x \in B(x^*,\eta),$$
$$x_N \in B(x^*,\eta/2),$$
$$|f(x_N)-f(x^*)|^{1-\theta} < \delta C(1-\theta)\eta/2.$$

If $f(x_n) \geq f(x^*)$ for all $n \in \mathbb{N}$, then $(x_n)_{n\geq N}$ lies entirely in $B(x^*,\eta)$.

ii. *Suppose that f is semialgebraic. Then if $(x_n)_{n\in\mathbb{N}}$ has a cluster point $x^* \in \mathbb{R}^d$, it converges to x^*.*

Proof.

i. By assumption, we have $x_N \in B(x^*,\eta/2)$ and

$$|f(x_N)-f(x^*)|^{1-\theta} < \delta C(1-\theta)\eta/2.$$

It follows from Lemma 5 that

$$\delta C(1-\theta)\|x_{N+1}-x_N\| \leq (f(x_N)-f(x^*))^{1-\theta} - (f(x_{N+1})-f(x^*))^{1-\theta}.$$

Let us prove by strong induction that for every $n \geq N$, $x_n \in B(x^*,\eta)$. Assume $n \geq N+1$ and suppose that for every integer $N \leq k \leq n-1$, $x_k \in B(x^*,\eta)$. Lemma 5 yields

$$(\forall\, k \in [N,\ldots,n-1]) \quad \delta C(1-\theta)\|x_{k+1}-x_k\| \leq (f(x_k)-f(x^*))^{1-\theta} - (f(x_{k+1})-f(x^*))^{1-\theta}.$$

Summing both sides over k yields

$$\begin{aligned}\delta C(1-\theta)\sum_{k=N}^{n-1}\|x_{k+1}-x_k\| &\leq (f(x_N)-f(x^*))^{1-\theta} - (f(x_n)-f(x^*))^{1-\theta}\\ &\leq (f(x_N)-f(x^*))^{1-\theta} < \delta C(1-\theta)\eta/2.\end{aligned} \tag{3}$$

Because

$$\|x_n-x^*\| \leq \sum_{k=N}^{n-1}\|x_{k+1}-x_k\| + \|x_N-x^*\| < \eta/2+\eta/2 = \eta,$$

we have $x_n \in B(x^*,\eta)$. We have proved thus that for every $n \geq N$, $x_n \in B(x^*,\eta)$.

ii. Since $(f(x_n))_{n\in\mathbb{N}}$ is nonincreasing by 1, we deduce that $f(x_n)$ converges and since x^* is a cluster point, $f(x_n) \to f(x^*)$ and $f(x_n) \geq f(x^*)$ for all $n \in \mathbb{N}$. Since f is semialgebraic, f has the Łojasiewicz property at x^* [24]. Hence, let us define θ, C, and η as in Definition 3 relative to x^*. Since x^* is a cluster point and $f(x_n) \to f(x^*)$, there exists N as in i above.

For every integer $n \geq N$, it follows from (3) that

$$\delta C(1-\theta)\sum_{k=N}^{n-1}\|x_{k+1}-x_k\| \leq (f(x_N)-f(x^*))^{1-\theta} \leq (f(x_0)-f(x^*))^{1-\theta} < +\infty,$$

and hence the series converges, the increments are summable, and $(x_k)_{k\in\mathbb{N}}$ converges to x^*. ◀

▶ Remark 7 (Convergence and semialgebraicity).

a. Note that when f is semialgebraic, Theorem 6 describe two exclusive alternatives, for any sequence:
- either $\|x_n\| \to \infty$
- or $(x_n)_{n\in\mathbb{N}}$ converges to a critical point x^*.

Indeed, if we are not in the diverging case, there is a cluster point x^* which must be a critical point. Whence we are in the situation of i above.

b. If f is, in addition, coercive, i.e., $\lim_{\|x\|\to+\infty} f(x) = +\infty$, each Hölder gradient sequence converges to a critical point since the first alternative is not possible because $(f(x_k))_{k\in\mathbb{N}}$ is non increasing so that $(x_k)_{k\in\mathbb{N}}$ is bounded.

▶ Remark 8 (Armijo line search). The two conditions in Theorem 6 are satisfied if one uses the classical Armijo backtracking line search strategy (see e.g. [7]). This strategy only uses C^1 smoothness of f. We adapt this strategy in further sections to take advantage of qualitative Hölderian regularity of ∇f.

2.3 Gradient descent for nonconvex functions with globally Hölderian gradient

To illustrate our recipe, we consider first the ideal case of a gradient method on a globally Hölder function with known constants [29, 39]. We study Algorithm 1, previously presented in [39] for which we prove sequential convergence.

Assumption 1 (Global Hölder regularity)**.** *The function $f\colon \mathbb{R}^d \to \mathbb{R}$ is C^1, semialgebraic, and*

$$\forall\, x_1, x_2 \in \mathbb{R}^d, \quad \|\nabla f(x_1) - \nabla f(x_2)\| \leq \beta\|x_1 - x_2\|^\nu, \text{ with } \beta > 0,\ \nu \in]0,1]. \tag{4}$$

Algorithm 1: Hölder gradient method

Input: $\nu \in]0,1]$, $\beta \in]0,+\infty[$ and $\gamma \in]0,(\nu+1)/\beta[$
Initialization: $x_0 \in \mathbb{R}^d$
for $n = 0, 1, \ldots$ **do**
$\gamma_n(x_n) = \gamma\left(\frac{\nu+1}{\beta}\right)^{1/\nu-1}\|\nabla f(x_n)\|^{1/\nu-1}$
$x_{n+1} = x_n - \gamma_n(x_n)\nabla f(x_n)$

Proposition 9 (Convergence of the Hölder gradient method for nonconvex functions)**.** *Under Assumption 1, consider a sequence $(x_n)_{n\in\mathbb{N}}$ generated by Algorithm 1. Then the following hold:*

1. *the sequence $(f(x_n))_{n\in\mathbb{N}}$ is nonincreasing,*
2. *if the sequence $(x_n)_{n\in\mathbb{N}}$ has a cluster point, then it converges to a critical point $x^* \in \mathbb{R}^d$ of f, i.e., $\nabla f(x^*) = 0$,*
3. *under the assumptions of 2, for every [3] $n \in \mathbb{N}$,*

$$\min_{0\leq k\leq n}\|\nabla f(x_k)\|^{\frac{1}{\nu}+1} \leq \left[\frac{f(x_0)-f(x^*)}{\gamma-\gamma^{\nu+1}\left(\frac{\beta}{\nu+1}\right)^\nu}\left(\frac{\beta}{\nu+1}\right)^{\frac{1-\nu}{\nu}}\right]\frac{1}{n+1} = O\left(\frac{1}{n}\right).$$

Choosing $\gamma = \frac{\nu+1}{\beta}\left(\frac{1}{\nu+1}\right)^{1/\nu}$, we obtain

$$\min_{0\leq k\leq n}\|\nabla f(x_k)\|^{\frac{1}{\nu}+1} \leq \frac{(f(x)-f(x^*))\beta^{1/\nu}(\nu+1)}{\nu(n+1)}.$$

[3] This result is essentially present in [39]

Proof. Let $n \in \mathbb{N}$ and set $d_n = \nabla f(x_n)$. For the clarity of the proof, the dependence of γ_n in x_n is dropped. Lemma 4 with $U = \mathbb{R}^d$ provides

$$\begin{aligned} f(x_{n+1}) &\leq f(x_n) + \langle d_n, x_{n+1} - x_n \rangle + \frac{\beta}{\nu+1}\|x_{n+1} - x_n\|^{\nu+1} \\ &\leq f(x_n) - \frac{1}{\gamma_n}\|x_{n+1} - x_n\|^2 + \frac{\beta}{\nu+1}\|x_{n+1} - x_n\|^{\nu+1} \\ &= f(x_n) - \frac{1}{\gamma_n}\big(\|x_{n+1} - x_n\|^2 - \frac{\beta\gamma_n}{\nu+1}\|x_{n+1} - x_n\|^{\nu+1}\big). \end{aligned} \tag{5}$$

By definition of γ_n we have

$$\gamma_n^{1/\nu} = \gamma \left(\frac{\nu+1}{\beta}\right)^{1/\nu-1} \|x_{n+1} - x_n\|^{1/\nu-1}$$

and thus

$$\gamma_n = \gamma^\nu \left(\frac{\nu+1}{\beta}\right)^{1-\nu} \|x_{n+1} - x_n\|^{1-\nu}.$$

Set $\delta = 1 - \gamma^\nu \left(\frac{\beta}{\nu+1}\right)^\nu$. Since $\gamma < \frac{\nu+1}{\beta}$ by hypothesis in Algorithm 1, we have $\delta > 0$ and we deduce from (5) that

$$\begin{aligned} f(x_{n+1}) &\leq f(x_n) - \frac{1}{\gamma_n}\left(\|x_{n+1} - x_n\|^2 - \gamma^\nu \left(\frac{\beta}{\nu+1}\right)^\nu \|x_{n+1} - x_n\|^2\right) \\ &= f(x_n) - \frac{1 - \gamma^\nu \left(\frac{\beta}{\nu+1}\right)^\nu}{\gamma_n}\|x_{n+1} - x_n\|^2 \\ &= f(x_n) - \frac{\delta}{\gamma_n}\|x_{n+1} - x_n\|^2. \end{aligned} \tag{6}$$

Hence $(f(x_n))_{n\in\mathbb{N}}$ is nonincreasing, this proves 1. Since, for all n, $\|\nabla f(x_n)\| = \|x_{n+1} - x_n\|/\gamma_n$ and $(x_n)_{n\in\mathbb{N}}$ has a cluster point, we can apply Theorem 6 and obtain that $x_n \to x^* \in \mathbb{R}^d$. Finally, it follows from (6) that $\|x_{n+1} - x_n\| \to 0$ and that $\|x_{n+1} - x_n\| = \gamma_n\|\nabla f(x_n)\| = \gamma \left(\frac{\nu+1}{\beta}\right)^{1/\nu-1} \|\nabla f(x_n)\|^{1/\nu} \to 0$. Hence x^* is a critical point, which proves 2.

For n fixed, we have using the definition of γ_n,

$$\frac{\delta}{\gamma_n}\|x_{n+1} - x_n\|^2 = \delta\gamma_n\|\nabla f(x_n)\|^2 = \delta\gamma \left(\frac{\nu+1}{\beta}\right)^{\frac{1}{\nu}-1} \|\nabla f(x_n)\|^{\frac{1}{\nu}+1}.$$

Then it follows from (6) that

$$\|\nabla f(x_n)\|^{\frac{1}{\nu}+1} \leq \frac{1}{\delta\gamma}\left(\frac{\beta}{\nu+1}\right)^{\frac{1-\nu}{\nu}} (f(x_n) - f(x_{n+1})) \tag{7}$$

whence

$$\begin{aligned} (n+1)\min_{k=0,\dots,n}\|\nabla f(x_k)\|^{\frac{1}{\nu}+1} &\leq \sum_{k=0}^{n}\|\nabla f(x_k)\|^{\frac{1}{\nu}+1} \\ &\leq \frac{1}{\gamma - \gamma^{\nu+1}\left(\frac{\beta}{\nu+1}\right)^\nu}\left(\frac{\beta}{\nu+1}\right)^{\frac{1-\nu}{\nu}} (f(x_0) - f(x^*)). \end{aligned}$$

Choosing $\gamma = \frac{\nu+1}{\beta}\left(\frac{1}{\nu+1}\right)^{1/\nu}$, we obtain

$$\begin{aligned} \gamma - \gamma^{\nu+1}\left(\frac{\beta}{\nu+1}\right)^\nu &= \frac{\nu+1}{\beta}\left(\frac{1}{\nu+1}\right)^{1/\nu} - \left(\frac{\nu+1}{\beta}\left(\frac{1}{\nu+1}\right)^{1/\nu}\right)^{\nu+1}\left(\frac{\beta}{\nu+1}\right)^\nu \\ &= \frac{\nu+1}{\beta}\left(\frac{1}{\nu+1}\right)^{1/\nu}\left(1 - \frac{1}{1+\nu}\right) \\ &= \frac{\nu+1}{\beta}\left(\frac{1}{\nu+1}\right)^{1/\nu}\left(\frac{\nu}{1+\nu}\right), \end{aligned}$$

from which we deduce

$$(n+1)\min_{0\le k\le n}\|\nabla f(x_k)\|^{\frac{1}{\nu}+1} \le \frac{(f(x)-f(x^*))(\nu+1)}{\nu}\frac{\left(\frac{\beta}{\nu+1}\right)^{\frac{1-\nu}{\nu}}}{\frac{\nu+1}{\beta}\left(\frac{1}{\nu+1}\right)^{1/\nu}}$$
$$= (f(x)-f(x^*))\beta^{1/\nu}\frac{\nu+1}{\nu},$$

which proves 3. ◀

3 The "Backtrack Hölder" gradient algorithm and diagonal backtracking

In practice, the constants are unknown and the Hölderian properties are merely local. The algorithm we present now (Algorithm 2), is in the spirit of the classical backtracking method, see e.g., [7]. The major difference with classical approaches, is that we devise a *diagonal backtracking*, to detect both constants β, ν at once, in a single searching pass. The term *diagonal* refers to the idea that the two parameters describing Hölder regularity of the gradient vary together during the search process.

Assumption 2. *The function $f\colon \mathbb{R}^d \to \mathbb{R}$ is a C^1 semialgebraic*[4]*.*

This implies that ∇f is locally Hölder by Proposition 2.

In the following algorithm, $\alpha, \gamma > 0$ are step length parameters, $\delta > 0$ is a sufficient-decrease threshold and $\rho > 0$ balances the search between the unknown exponent ν and the unknown multiplicative constant β, see Assumption 1.

Algorithm 2: Backtrack Hölder gradient method

Input: $\delta, \alpha \in]0,1[$ and $\gamma, \rho \in]0,+\infty[$
Initialization: $x_0 \in \mathbb{R}^d$, $k_{-1} = 0$
for $n = 0, 1, \dots$ **do**
  $k = k_{n-1}$
  $\gamma_n(x_n) = \alpha^k \min\{1, \|\nabla f(x_n)\|^{\rho k}\}\gamma$
  while $f(x_n - \gamma_n(x_n)\nabla f(x_n)) > f(x_n) - \delta\gamma_n(x_n)\|\nabla f(x_n)\|^2$ **do**
    $k = k + 1$
    $\gamma_n(x_n) = \alpha^k \min\{1, \|\nabla f(x_n)\|^{\rho k}\}\gamma$
  $k_n = k$
  $x_{n+1} = x_n - \gamma_n(x_n)\nabla f(x_n)$

The following theorem provides convergence guarantees under local Hölder continuity (Assumption 2 for Algorithm 2). The first three items only use C^1 smoothness of f and the quantitative decrease ensured by line search in order to apply Theorem 6, while the quantitative estimates rely on the Hölderian assumption and the specific form of the line search strategy.

Theorem 10 (Convergence of Backtrack Hölder for nonconvex functions)**.** *Under Assumption 2, consider a sequence $(x_n)_{n\in\mathbb{N}}$ generated by Algorithm 2. Then the following hold:*

1. *$(\gamma_n)_{n\in\mathbb{N}}$ is well defined,*
2. *the sequence $(f(x_n))_{n\in\mathbb{N}}$ is nonincreasing,*
3. *if $(x_n)_{n\in\mathbb{N}}$ has a cluster point, then there exists $x^* \in \mathbb{R}^d$ such that $x_n \to x^*$ and $\nabla f(x^*) = 0$,*
4. *if $(x_n)_{n\in\mathbb{N}}$ has a cluster point, then the while-loop has a uniform finite bound $\bar{k} := \sup_{n\in\mathbb{N}} k_n < +\infty$. Moreover*
$$\min_{0\le i\le n}\|\nabla f(x_i)\| = O\left(\frac{1}{n^{\frac{1}{2+\rho\bar{k}}}}\right).$$
5. *Suppose, in addition to 4, that there exist $\beta \in]0,+\infty[$ and $\nu \in]0,1]$, such that ∇f is globally (β,ν) Hölder. Then*
$$\sup_{n\in\mathbb{N}} k_n \le 1 + \frac{1}{\nu}\max\left\{\frac{\log\left(\frac{(1-\delta)(\nu+1)}{\gamma^\nu\beta}\right)}{\log(\alpha)}, \frac{1-\nu}{\rho}\right\}. \tag{8}$$

[4] The assumption could simply be that, on bounded sets, the restriction of f belongs to a polynomially bounded o-minimal structure.

Proof.

1. For every $n \in \mathbb{N}$, we need to test for the existence of $\widetilde{\gamma} > 0$, such that

$$f(x_n - \widetilde{\gamma}\nabla f(x_n)) \leq f(x_n) - \delta\widetilde{\gamma}\|\nabla f(x_n)\|^2.$$

This test is obviously satisfied if $\nabla f(x_n) = 0$. Assuming the contrary, by using Taylor's expansion on f, we have as $\widetilde{\gamma} \to 0$

$$\begin{aligned} &f(x_n - \widetilde{\gamma}\nabla f(x_n)) \\ &= f(x_n) - \widetilde{\gamma}\|\nabla f(x_n)\|^2 + o(\widetilde{\gamma}) = f(x_n) - \delta\widetilde{\gamma}\|\nabla f(x_n)\|^2 + o(\widetilde{\gamma}) - \widetilde{\gamma}(1-\delta)\|\nabla f(x_n)\|^2. \end{aligned}$$

Since $\delta < 1$, for all small $\widetilde{\gamma}$, the right hand side is smaller than $f(x_n) - \delta\widetilde{\gamma}\|\nabla f(x_n)\|^2$. This shows that the test is satisfied by sufficiently small $\widetilde{\gamma}$ and the while loop has to terminate.

2. It follows from Algorithm 2 that for every $n \in \mathbb{N}$,

$$f(x_{n+1}) \leq f(x_n) - \frac{\delta}{\gamma_n}\|x_{n+1} - x_n\|^2, \tag{9}$$

and hence the descent property holds.

3. One has $\|\nabla f(x_n)\| = \frac{1}{\gamma_n}\|x_{n+1} - x_n\|$. Since $(x_n)_{n\in\mathbb{N}}$ has a cluster point, we conclude by Theorem 6 ii that there exists $x^* \in \mathbb{R}^d$ such that $x_n \to x^*$. This follows directly from 2 and (9).

4. Since f is locally Hölder and $x_n \to x^*$ as $n \to \infty$, there exist $U \subset \mathbb{R}^d$, a convex neighborhood of x^*, $\nu \in]0,1]$, and $\beta \in]0, +\infty[$ such that ∇f is (β, ν) Hölder on U and $(x_n)_{n\geq N}$ remains in U for N sufficiently large.

Fix any $K \in \mathbb{N}$ such that

$$K \geq \max\left\{\frac{\log\left(\frac{(1-\delta)(\nu+1)}{\gamma^\nu\beta}\right)}{\log(\alpha)\nu}, \frac{1-\nu}{\rho\nu}\right\}. \tag{10}$$

Then we also have

$$\alpha^K \leq \frac{1}{\gamma}\left(\frac{(1-\delta)(\nu+1)}{\beta}\right)^{1/\nu} \text{ and } \quad \rho K \geq \frac{1}{\nu} - 1.$$

Choosing any $\lambda = \alpha^K \min\{1, \|\nabla f(x)\|^{\rho K}\}\gamma$, we deduce that for any $x \in U$ such that $x - \lambda\nabla f(x) \in U$,

$$\begin{aligned} \lambda^\nu = \alpha^{K\nu}\min\{1, \|\nabla f(x)\|^{\rho K\nu}\}\gamma^\nu &\leq \left(\frac{(1-\delta)(\nu+1)}{\beta}\right)\min\{1, \|\nabla f(x)\|^{\rho K\nu}\} \\ &\leq \left(\frac{(1-\delta)(\nu+1)}{\beta}\right)\min\{1, \|\nabla f(x)\|^{1-\nu}\}. \end{aligned} \tag{11}$$

We derive from Lemma 4 and (11) that for any $x \in U$

$$\begin{aligned} f(x - \lambda\nabla f(x)) &\leq f(x) - \lambda\|\nabla f(x)\|^2 + \frac{\beta}{\nu+1}\lambda^{\nu+1}\|\nabla f(x)\|^{\nu+1} \\ &\leq f(x) - \lambda\|\nabla f(x)\|^2 + \frac{\beta\lambda}{\nu+1}\frac{(1-\delta)(\nu+1)}{\beta}\min\{1, \|\nabla f(x)\|^{1-\nu}\}\|\nabla f(x)\|^{\nu+1}. \end{aligned}$$

Since $\min\{1, \|\nabla f(x)\|^{1-\nu}\}\|\nabla f(x)\|^{\nu+1} \leq \|\nabla f(x)\|^2$, we have for all $x \in U$ such that $x - \lambda\nabla f(x) \in U$,

$$\begin{aligned} f(x - \lambda\nabla f(x)) &\leq f(x) - \lambda\|\nabla f(x)\|^2 + (1-\delta)\lambda\|\nabla f(x)\|^2 \\ &= f(x) - \delta\lambda\|\nabla f(x)\|^2. \end{aligned} \tag{12}$$

Fix any $N_0 \in \mathbb{N}$ large enough such that $x_N \in U$ for all $N \geq N_0$. Suppose that $K = k_{N_0}$ satisfies (10), then for all $N \geq N_0$ we may consider equation (12) with $x = x_N$, $\lambda = \gamma_N$, noting that $x_{N+1} = x_N - \gamma_N\nabla f(x_N) \in U$. This is exactly the negation of the condition to enter the while-loop of Algorithm 2. Hence, by a simple recursion, the algorithm never enters the while-loop after step N_0 and we have $k_N = k_{N_0}$ for all $N \geq N_0$. On the other hand, if

k_{N_0} does not satisfy (10), then since k is incremented by 1 at each execution of the while loop, using the fact that (10) implies (12), it must hold that

$$k_N \leq 1 + \max\left\{\frac{\log\left(\frac{(1-\delta)(\nu+1)}{\gamma^\nu \beta}\right)}{\log(\alpha)\nu}, \frac{1-\nu}{\rho\nu}\right\}.$$

In all cases, it follows from the monotonicity of k_N in N that for all $N \in \mathbb{N}$,

$$k_N \leq 1 + \max\left\{k_{N_0}, \frac{\log\left(\frac{(1-\delta)(\nu+1)}{\gamma^\nu \beta}\right)}{\log(\alpha)\nu}, \frac{1-\nu}{\rho\nu}\right\}, \tag{13}$$

hence $(k_n)_{n\in\mathbb{N}}$ is bounded.

Now we use (9) and 2 which ensures that

$$\frac{\|x_{n+1}-x_n\|^2}{\gamma_n} = \gamma_n\|\nabla f(x_n)\|^2 = \alpha^{k_n}\min\{\|\nabla f(x_n)\|^2, \|\nabla f(x_n)\|^{\rho k_n+2}\}\gamma$$

is summable and thus tends to 0 as $n \to \infty$. Using the fact that $(k_n)_{n\in\mathbb{N}}$ is bounded, in any case we have, $\nabla f(x_n) \to 0$ as $n \to \infty$.

It follows for n large enough that $\|\nabla f(x_n)\| \leq 1$, from the while loop condition and the fact that $\bar{k} := \sup_{n\in\mathbb{N}} k_n < +\infty$, that

$$\delta\alpha^{\bar{k}}\|\nabla f(x_n)\|^{2+\rho\bar{k}}\gamma \leq \delta\alpha^{k_n}\|\nabla f(x_n)\|^{2+\rho k_n}\gamma = \delta\gamma_n(x_n)\|\nabla f(x_n)\|^2 \leq f(x_n) - f(x_{n+1}).$$

Using the convergence of $(f(x_n))_{n\in\mathbb{N}}$, and by summing the previous equation and taking the minimum, we obtain that $\min_{0\leq i\leq n}\|\nabla f(x_i)\|^{2+\rho\bar{k}} = O(1/n)$.

5. The result follows from (13) with $k_{N_0} = 0$, since in this case the same reasoning can be applied for all $N \in \mathbb{N}$ with $U = \mathbb{R}^d$. ◀

▶ Remark 11 (Diagonal backtracking alternatives and comments). In the previous theorem, $(k_n)_{n\in\mathbb{N}}$ is required to be a nondecreasing sequence and (8) is actually a bound on the total number of additional calls to the function in the while-loop. In practice, this approach might be too conservative and other strategies may provide much more aggressive steps at the cost of additional calls to the function. We will use two variations to update $(k_n)_{n\in\mathbb{N}}$:

- Initialize k to 0 for fine tuning to the price of longer inner loops (see Algorithm 8).
- For some iterations, decrease the value of k by 1 (see Algorithm 5 for example).

In Theorem 105, we need to suppose that the gradient is *globally* Hölder contrary to Assumption 2. Note that the cluster point assumption on the sequence $(x_n)_{n\in\mathbb{N}}$ in Proposition 9 and Theorem 10 are automatically satisfied if f has compact sublevel sets thanks to the decrease property.

4 Backtrack Hölder for min-max problems

Our method and proofs are presented in the context of solving min-max problems, but the techniques are identical for the min-min case. We assume that $\mathbb{R}^d, \mathbb{R}^{d'}$ are endowed with their Euclidean structure.

4.1 Framework: semialgebraicity and a single valued best response

Let $\mathcal{Y} \subset \mathbb{R}^{d'}$ be a nonempty closed semialgebraic set, see Definition 1.

Properties of the value function and its best response

Assumption 3 (Standing assumptions)**.** *The function L is a C^1 semialgebraic function on $\mathbb{R}^d \times \mathbb{R}^{d'}$ and $(x,y) \mapsto \nabla_x L(x,y)$ is jointly continuous. Furthermore, for any compact sets $K_1 \subset \mathbb{R}^d$ and $K_2 \subset \mathbb{R}^{d'}$, there exist $\beta_1, \beta_2 \in]0,+\infty[$ such that, $\forall\, x_1, x_2 \in K_1$, $\forall\, y_1, y_2 \in K_2$,*

$$\|\nabla_x L(x_1,y_1) - \nabla_x L(x_2,y_2)\| \leq \beta_1\|x_1 - x_2\| + \beta_2\|y_1 - y_2\|. \tag{14}$$

Borrowing the terminology from game theory, one defines the *value function* as $g(\cdot) = \max_{y\in\mathcal{Y}} L(\cdot, y)$ and the *best response mapping* $p(\cdot) = \operatorname{argmax}_{y\in\mathcal{Y}} L(\cdot, y)$ for $x \in \mathbb{R}^d$.

Assumption 4 (Well posedness)**.**
H1. *The response map $p(x)$ is nonempty and single valued for every $x \in \mathbb{R}^d$,*
H2. *p is continuous.*

The first part of the assumption is satisfied whenever $L(x, \cdot)$ is strictly concave, see e.g. [26]. Note also that if $L(x, \cdot)$ is concave, as in a dual optimal transport formulation, some regularization techniques can be used to obtain uniqueness and preserve semialgebraicity, see e.g., [15]. The continuity assumption H2, it is much less stringent than it may look as evidenced by the following result:

Proposition 12 (Continuity of p)**.** *Suppose that Assumption 3 and Assumption 4-H1 are satisfied, and that either $\mathcal{Y}$ is compact, or p is bounded on bounded sets. Then the best response p is a continuous function, that is Assumption 4-H2 is fulfilled.*

Proof. Let us proceed with the case when $\mathcal{Y}$ is compact; the other case is similar. Let $(x_n)_{n\in\mathbb{N}}$ be a sequence such that $x_n \to x^* \in \mathbb{R}^d$. We need to prove that $p(x_n) \to p\ (x^*)$. For every $n \in \mathbb{N}$, set $y_n = p\ (x_n)$ and since $\mathcal{Y}$ is compact, let $y^* \in \mathcal{Y}$ a cluster point of $(y_n)_{n\in\mathbb{N}}$. Since g and L are continuous we have $g(x_n) \to g(x^*)$ and $L(x_{n_k}, y_{n_k}) \to L(x^*, y^*)$. Since $p(x_{n_k}) = y_{n_k}$ one has $L(x_{n_k}, y_{n_k}) \geq L(x, y_{n_k})$ for all x in $\mathbb{R}^d$. Thus at the limit $L(x^*, y^*) \leq L(x, y^*)$ for all x in $\mathbb{R}^d$. This implies that $L(x^*, y^*) = g(x^*)$, and so, by uniquennes of the argmax, $p(x^*) = y*$. Whence p is continuous. ◀

Combining these assumptions with Tarski–Seidenberg theorem and the properties of semialgebraic functions [8], we obtain the following.

Proposition 13 (Properties of p and g)**.** *Suppose that Assumption 3 and Assumption 4 are satisfied. Then*
1. *g is differentiable and for all $\bar{x} \in \mathbb{R}^d$, $\nabla g(\bar{x}) = \nabla_x L(\bar{x}, p(\bar{x}))$,*
2. *both the value function g and the best response p are semialgebraic,*
3. *the gradient of the value function, ∇g, is locally Hölder.*

Proof.

1. This is a consequence of [33, Theorem 10.31].

2. According to the definition of a semi-agebraic function, we need to prove that their graph is semialgebraic. For g:

$$\operatorname{epi} g = \big\{(x,\xi) \in \mathbb{R}^d \times \mathbb{R} \;\big|\; g(x) \leq \xi\big\} = \big\{(x,\xi) \in \mathbb{R}^d \times \mathbb{R} \;\big|\; (\forall\, y \in \mathcal{Y}) \quad L(x,y) \leq \xi\big\},$$

and its complement set is $\big\{(x,\xi) \in \mathbb{R}^d \times \mathbb{R} \;\big|\; (\exists\, y \in \mathcal{Y})\ L(x,y) > \xi\big\}$ which is the projection of

$$\big\{(x,\xi,y) \in \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^{d'} \;\big|\; L(x,y) > \xi\big\} \bigcap \mathbb{R}^d \times \mathbb{R} \times \mathcal{Y}.$$

As a conclusion it is semialgebraic by Tarski–Seidenberg principle. The same being true for the hypograph, g is semialgebraic.

For p: $\operatorname{graph} p = \big\{(x,y) \in \mathbb{R}^d \times \mathcal{Y} \;\big|\; (\forall\, y' \in \mathcal{Y})\ L(x,y) \geq L(x,y')\big\}$. Then $\operatorname{graph} p$ is defined from a first-order formula and the conclusion follows from [14, Theorem 2.6].

3. Using Assumption 4-H2 and 2, p is continuous and semialgebraic so using Proposition 2, p is locally Hölder. Similarly Assumption 4-H2, 1 and 2 ensure that ∇g is also continuous and semialgebraic and the result follows again from Proposition 2. ◀

▶ Remark 14. Consider $L(x,y) = xy$, $\mathcal{Y} = [-1,1]$, one sees that $g(x) = \max_{y\in[-1,1]} L(x,y) = |x|$ while $p(x) = \operatorname{sign} x$ if $x \neq 0$ and $p(0) = [-1,1]$. This shows that Assumption 4 is a necessary assumption for g to be differentiable. One cannot hope in general for ∇g to be locally Lipschitz continuous. For instance set $\mathcal{Y} = \mathbb{R}_+$, $L(x,y) = xy - \frac{1}{3}y^3$, then $g(x) = \max_{y\in\mathbb{R}^+} L(x,y) = \frac{2}{3}x^{3/2}$ with $\nabla g(x) = \sqrt{x}$.

Comments and rationale of the method

At this stage the principles of our strategy can be made precise. We deal with problems that are unsymmetric in structure: the argmax is easily computable or approximable while the block involving the minimizing variable is difficult to handle. This suggests to proceed as follows: one computes a best response mapping, the gradient of the value function becomes accessible via formula 1 in Proposition 13, and thus a descent step can be taken. The questions are: *which steps are acceptable? Can they be tuned automatically?* This is the object of the next sections.

Gathering the results provided in Section 3, we provide here our main algorithm (Algorithm 3).

Algorithm 3: Monotone Backtrack Hölder for min-max

Input: $\delta, \alpha \in]0,1[$ and $\gamma, \rho \in]0,+\infty[$

Initialization: $x_0 \in \mathbb{R}^d$, $y_0 = \arg\max_{z\in\mathcal{Y}} L(x_0, z)$, $k_{-1} = 0$

for $n = 0, 1, \ldots$ **do**

$k = k_{n-1}$

$\gamma_n(x_n) = \gamma\alpha^k \min\{1, \|\nabla_x L(x_n, y_n)\|^{\rho k}\}$

$x = x_n - \gamma_n(x_n)\nabla_x L(x_n, y_n)$

$y = \arg\max_{z\in\mathcal{Y}} L(x, z)$

while $L(x, y) > L(x_n, y_n) - \delta\gamma_n(x_n)\|\nabla_x L(x_n, y_n)\|^2$ **do**

$k = k + 1$

$\gamma_n(x_n) = \gamma\alpha^k \min\{1, \|\nabla_x L(x_n, y_n)\|^{\rho k}\}$

$x = x_n - \gamma_n(x_n)\nabla_x L(x_n, y_n)$

$y = \arg\max_{z\in\mathcal{X}} L(x, z)$

$k_n = k, \quad x_{n+1} = x, \quad y_{n+1} = y.$

Several comments are in order:

- Algorithm 3 contains an inner loop whose overhead cost becomes negligible as $n \to \infty$, which allows for automatic step size tuning. The form of Algorithm 3 is slightly different from Algorithm 2 to avoid duplicate calls to the max-oracle required both to compute gradients and evaluate functions.
- As described in Remark 11, the backtracking strategy is one among others and it is adaptable to different settings. In this min-max case, the cost of the max-oracle may have some impact: either it is costly and extra-flexibility is needed or it is cheap and it can be kept as is. Two examples are provided in Sections 5.1 and 5.2.
- A direct modification of the above method provides also an algorithm for

$$\min_{x\in\mathbb{R}^d} \min_{y\in\mathcal{Y}} L(x, y). \tag{15}$$

- Algorithm 3 is a model algorithm corresponding to a monotone backtracking approach (i.e., the sequence (k_n) is nondecreasing), but many other variants are possible; see Section 5.3. Algorithms 5 and 6 are for the min-min problem, with a non monotone backtracking and the same guarantees. A heuristic version is also considered: it is Algorithm 8 where an approximation of the argmax is used.

To benchmark our algorithms, we compare them to Algorithms 4, 6, and 7 in Section 5.3, with constant but finely tuned step sizes or with Armijo search.

Theorem 15 (Backtrack Hölder for min-max)**.** *Under Assumptions 3 and 4, consider the sequences $(x_n)_{n\in\mathbb{N}}$ and $(y_n)_{n\in\mathbb{N}}$ generated by Algorithm 3. Suppose that $(x_n)_{n\in\mathbb{N}}$ has a cluster point. Then*

1. *The while-loop has a uniform bound, i.e., $\sup_{n\in\mathbb{N}} k_n < +\infty$.*
2. *$(x_n)_{n\in\mathbb{N}}$ converges to x^* in $\mathbb{R}^d$ and $(y_n)_{n\in\mathbb{N}}$ converges to $y^* \in \mathcal{Y}$, with $\nabla_x L(x^*, y^*) = 0$ and $y^* = \operatorname{argmax}_{y\in\mathcal{Y}} L(x^*, y)$.*
3. *Suppose that there exist $\beta \in]0,+\infty[$ and $\nu \in]0,1]$ such that ∇g is (β, ν) Hölder everywhere. Then the cost of the while-loop is bounded by*

$$\sup_{n\in\mathbb{N}} k_n \leq 1 + \frac{1}{\nu} \max\left\{ \frac{\log\left(\frac{(1-\delta)(\nu+1)}{\gamma^\nu \beta}\right)}{\log(\alpha)}, \frac{1-\nu}{\rho} \right\}. \tag{16}$$

Proof. Recall that $g(\cdot) = \max_{y\in\mathcal{Y}} L(\cdot, y)$ and $p(\cdot) = \operatorname{argmax}_{y\in\mathcal{Y}} L(\cdot, y)$. It follows from Proposition 13 that for every $n \in \mathbb{N}$, $\nabla_x L(x_n, y_n) = \nabla g(x_n)$. We derive from Proposition 3 that ∇g is locally Hölder. It turns out that Algorithm 3 applied to L is the same as Algorithm 2 applied to g. Thus Theorem 10 ensures the convergence of $(x_n)_{n\in\mathbb{N}}$ to a critical point $x^* \in \mathbb{R}^d$ of g. Furthermore, it follows from the continuity of p that $y_n \to y^* = p\,(x^*)$. We conclude that $(x_n, y_n)_{n\in\mathbb{N}}$ converges to a critical point of L, satisfying $y^* = \operatorname{argmax}_{y\in\mathcal{Y}} L(x^*, y)$. Finally, since for every $n \in \mathbb{N}$, $\nabla g(x_n) = \nabla_x L(x_n, y_n)$, we conclude by Theorem 105. ◀

▶ Remark 16. In [19, Proposition 2], the authors mention an algorithm akin to what we proposed, but without backtracking. They mention the fact that if one had access to the max-oracle, then one would be able to implement a gradient descent by using "sufficiently small updates". Our theoretical results are an answer to this comment as we offer a quantitative characterization of how small the steps should be, as well as a backtracking estimation technique.

5 Numerical experiments

We compare our method with constant step size algorithm and Armijo backtracking for the Generative Adversarial Network (GAN) problem, first using Sinkhorn divergences and second considering Wasserstein adversarial networks. Data lie in $\mathbb{R}^d = \mathbb{R}^2$, the sample size is $N = 1024$ and we consider $x_1, \dots, x_N \in \mathbb{R}^d$ a fixed sample from a distribution $\mathbb{P}_d$, which is a Gaussian mixture, see Figure 1, and $z_1, \dots, z_N \in \mathcal{Z}$ a fixed sample from latent distribution $\mathbb{P}_z = U([0,1]\times[0,1])$, uniform on $\mathcal{Z}$, where $\mathcal{Z} = [0,1]\times[0,1]$.

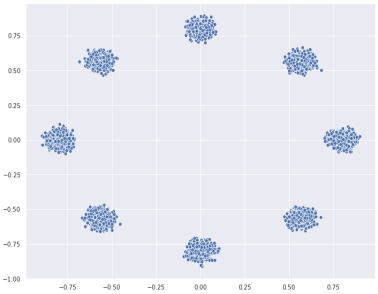

Figure 1 Data distribution $x_1, \dots, x_N$

We consider for the *generator* G, a dense neural network with three hidden layers containing respectively 64, 32, 16 neurons with a ReLU activation between each layer. We write $G : \mathcal{Z}\times\Theta_G \to \mathcal{X}$, with inputs in $\mathcal{Z}$ and parameters $\theta_G \in \Theta_G$ where $\Theta_G = \mathbb{R}^q$ with q the total number of parameters of the network (2834 in our case).

5.1 Sinkhorn GAN

We first consider training generative a network using Sinkhorn divergences as proposed in [17]. This is a min-min problem which satisfies Assumption 4, except for semialgebraicity (see also the remark in Equation 15). As detailed in Remark 17 even if the semialgebraic assumption does not hold, the local Hölderian properties which we require still hold which is sufficient to apply the results of Theorem 10 and Proposition 9. Sinkhorn algorithm [15, 35] allows us to compute a very precise approximation of the min-oracle required by our algorithm, we use it as an exact estimate. Note that the transport plan for the Sinkhorn divergence is regularized by an entropy term whence the inner minimization problem has a unique solution and the corresponding p is continuous. This is a perfect example to illustrate our ideas. Consider the following probability measures

$$\bar{\mu} = \frac{1}{N}\sum_{i=1}^{N} \delta_{x_i}, \qquad \text{(empirical target distribution)},$$

$$\mu(\theta_G) = \frac{1}{N}\sum_{i=1}^{N} \delta_{G(z_i,\theta_G)}, \qquad \text{(empirical generator distribution)}.$$

We then define the *Sinkhorn divergence* between these two distributions as

$$\mathcal{W}_\epsilon(\bar{\mu}, \mu(\theta_G)) = \min_{P\in\mathbb{R}_+^{N\times N}} \left\{ \operatorname{Tr}(PC(\theta_G)^T) + \epsilon \sum_{i,j=1}^{N} P_{ij}\log(P_{ij}) \;;\; P1_N = 1_N, P^T 1_N = 1_N \right\}$$

where $\epsilon > 0$ is a regularization parameter, $C(\theta_G) = [\|G(z_i, \theta_G) - x_j\|]_{i,j} \in \mathbb{R}^{N\times N}$ is the pairwise distance matrix between target observations $(x_i)_{1\leq i\leq N}$ and generated observations $(G(z_i, \theta_G))_{1\leq i\leq N}$. Here Tr is the trace, and 1_N is the all-ones vector. The optimum is unique thanks to the entropic regularization and the optimal transportation plan P can be efficiently estimated with an arbitrary precision by the Sinkhorn algorithm [15, 35].

Training our generative network amounts to solving the following min-min problem

$$\min_{\theta_G} \mathcal{W}_\epsilon(\bar{\mu}, \mu(\theta_G)). \tag{17}$$

▶ Remark 17 (Global subanalyticity ensures Łojasiewicz inequality). The cost function of the Sinkhorn GAN problem (17) is not semialgebraic due to the log function. However, we never use the logarithm in a neighborhood of 0 during the optimization process because of its infinite slope. Therefore the loss acts as a globally subanalytic function during the process. Hence p, g are globally subanalytic and the Łojasiewicz inequality as well as Hölder properties still hold, see [4, 8, 10] for more on this.

Algorithmic strategies for Sinkhorn GAN

The monotone diagonal backtracking is too conservative for this case, so we use a variant described in Algorithm 5 instead. At each step, the idea is to try to *decrement* k by 1 whenever possible, keeping some sufficient-decrease property valid. Otherwise k is increased as in the monotone method, until sufficient decrease of the value is ensured. This approach is particularly adapted to the Sinkhorn case, because estimating the best response is cheap.

Note that, to propose a fair comparison and keep the same complexity between algorithms, we count each call to the min-oracle, both in the outer and in the inner while loop, as an iteration step. The parameters used in this experiment for Algorithm 5 are $\gamma = 1$, $\alpha = 0.5$, $\delta = 0.25$, $\delta_+ = 0.95$, and $\rho = 0.5$. We compare with Algorithm 4, which is a constant step size variant, with different step-size parameters $\gamma \in \{0.01, 0.05, 0.1\}$. We compare with the standard Armijo backtracking algorithm (see Algorithm 6), which uses a similar approach as in Algorithm 5, to tune the step size parameter γ_n, but does not take advantage of the Hölder property. All algorithms are initialized randomly with the same seed.

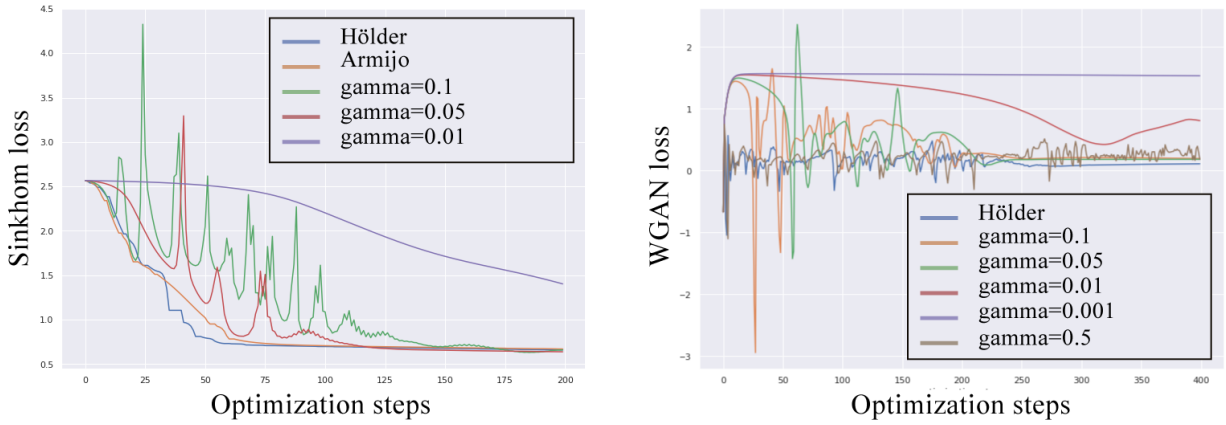

Figure 2 *Left:* Sinkhorn loss with respect to number of Sinkhorn max-oracle evaluation for different gradient step rules. The x axis accounts for all oracle calls, not only the ones used to actually perform gradient steps. "Hölder" and "Armijo", refer to the proposed Hölder backtracking and Armijo backtracking schemes described in Algorithm 5 and 6 while "gamma" referes to the constant step size variant in Algorithm 4. *Right:* WGAN loss with respect to iteration number for different gradient step rules. "Hölder" refers to the heuristic extension of the proposed method in Algorithm 8, while "gamma" referes to constant step sizes in Algorithm 7.

We observe on the left part of Figure 2 that both Hölder and Armijo backtracking provide decreasing sequence and avoid oscillations. Both algorithms converge faster than the constant step size variant. Furthermore, since our algorithm can take into account the norm of the gradient, the number of inner-loop iterations is smaller than Armijo backtracking, and that explain why the Non Monotone Hölder backtracking is faster.

5.2 Wasserstein GAN

We treat the Wasserstein GAN (WGAN) heuristically with an approximation of the max-oracle and use Algorithm 8, which matches this setting.

Consider a second neural network, called *discriminator*, $D : \mathbb{R}^d \times \Theta_D \to \mathbb{R}$ with inputs in $\mathbb{R}^d$ and parameters $\theta_D \in \Theta_D$ whose architecture is the same as G (i.e., $\Theta_D = \Theta_G$) but with a fullsort activation between each layer, see [2]. We consider the following problem

$$\min_{\theta_G} \max_{\theta_D} \sum_{i=1}^{n} D(x_i, \theta_D) - \sum_{j=1}^{n} D(G(z_j, \theta_G), \theta_D). \tag{18}$$

In order to implement the analogy with Kantorovitch duality in the context of GANs [3], one has to ensure that the discriminator D is 1-Lipschitz, when seen as a function of its input neurons. This is enforced using a specific architecture for the discrimintator network D. We use Bjork orthonormalization and fullsort activation functions [2], which ensure that the network is 1-Lipschitz without any restriction on its weight parameters θ_D.

For this problem, we use Algorithm 8, which is a heuristic modification of our method designed to deal with the inner max. Both the argmax and max are indeed approximated by gradient ascent. Algorithm 8 then implements the same bactracking idea which is evaluated on the benchmark of the previous section. Doing so, the extra-cost incurred by the while-loop becomes negligible and we can find the optimal value of k by exhaustive search. For this reason, in this heuristic context, Hölder backtracking schemes have very little advantage compared to Armijo and we do not report comparison. Detailed investigations for large scale networks is postponed to future research. Since GAN's training is delicate in practice [21], we provide comparison with many step size choices for the constant step size algorithm. As for Backtrack Hölder min-max, we use parameters $\gamma = 1$, $\delta = 0.75$, $\alpha = 0.75$, and $\rho = 0.20$ for the Hölder backtracking algorithm and constant step size parameter $\gamma \in \{0.001, 0.01, 0.05, 0.1, 0.5\}$ for the constant step size variant. All algorithms are initialized randomly with the same seed.

Figure 2 displays our results on the right. The optimal loss equals 0. One observes that constant large steps are extremely oscillatory while small steps are stable but extremely slow. Backtrack Hölder takes the best of the two world, oscillates much less and stabilizes with a smaller loss value compared to constant step size variants. In Algorithm 8 the partial maximum is only apporximated. In practice, the algorithm alternates between gradients steps on the maximization and minimization problems. Therefore, at early stages of the algorithm, one may find objective values below the optimal threshold because of the bad quality of the maximization process. For later iterations, the maximization process stabilizes and so does the min-max loss. This is an intrinsic difficulty in GAN problems where the size of the objects involved does not allow for exact evaluation of partial maximizers. The proposed experiment is a mere heuristic at this stage and the corresponding convergence analysis is a hard question which we leave for future research.

5.3 Numerical Experiments: technical complements

The experiments presented in Sections 5.1 & 5.2 do not directly relate to the min-max algorithms as presented in Section 4. Indeed, the Sinkhorn GAN problem uses a regularized primal formulation of optimal transport and is therefore written as a min-min problem. The arguments described in Section 4 apply *mutatis mutandis* to this setting. The benchmark corresponds to the following algorithms: constant step-size, Hölder backtracking and Armijo backtracking. Secondly, in the Wasserstein GAN example, the argmax cannot be evaluated exactly and is approximated by gradient ascent, an operation which is very costly. In this context, we propose a heuristic variation on our backtracking strategy which only require one approximate partial maximization at each step.

5.3.1 Sinkhorn GAN

Sinkhorn GAN is a min-min problem, thus our model must be slightly adapted. First, we start with Algorithm 4 below which is a constant step size algorithm. Due to the specific setting of Sinkhorn problem, the argmin may be computed exactly. The next algorithm is a Backtrack Hölder method for the min-min problem. For gaining efficiency, we introduce a new rule in Algorithm 5, which maintains the sufficient decrease property, without the monotonicity of $(k_n)_{n\in\mathbb{N}}$. This allows for a possible adaption to local smoothness, taking larger steps for some iterations. All the results of Theorem 15 apply to this algorithm when applied to the Sinkhorn GAN problem.

We also present an Armijo search process for this problem in Algorithm 6. It has a structure similar to the "Non Monotone Hölder Backtrack" but with a much less clever update for γ_n.

Algorithm 4: Constant step size gradient method for min-min

Input: $\gamma \in]0, +\infty[$
Initialization : $x_0 \in \mathbb{R}^d$
for $n = 0, 1, \dots$ **do**
  $y_n = \operatorname{argmin}_{y \in \mathcal{Y}} L(x_n, y)$
  $x_{n+1} = x_n - \gamma \nabla_x L(x_n, y_n)$

Algorithm 5: Non Monotone Backtrack Hölder for min-max

Input: $N \in \mathbb{N}$,$\gamma, \rho \in]0, +\infty[$, and $\alpha, \delta, \delta_+ \in]0, 1[$
Initialization : $k_{-1} = 1$, $n = 0$, and $x_0 \in \mathbb{R}^d$
for $n < N$ **do**
  $k = k_{n-1}$
  $\gamma_n(x_n) = \gamma \alpha^k \min\{1, \|\nabla_x L(x_n, y_n)\|^{\rho k}\}$
  $x = x_n - \gamma_n(x_n) \nabla_x L(x_n, y_n)$
  $y = \arg\min_{z \in \mathcal{Y}} L(x, z)$
  if $L(x, y) < L(x_n, y_n) - \delta^+ \gamma_n(x_n) \|\nabla f(x_n)\|^2$ **then**
    $k = k - 1$
  while $L(x, y) > L(x_n, y_n) - \delta \gamma_n(x_n) \|\nabla_x L(x_n, y_n)\|^2$ **do**
    $k = k + 1$
    $\gamma_n(x_n) = \gamma \alpha^k \min\{1, \|\nabla_x L(x_n, y_n)\|^{\rho k}\}$
    $x = x_n - \gamma_n(x_n) \nabla_x L(x_n, y_n)$
    $y = \arg\min_{z \in \mathcal{X}} L(x, z)$
  $k_n = k, \quad x_{n+1} = x, \quad y_{n+1} = y.$

Algorithm 6: Non Monotone Armijo for min-max

Input: $N \in \mathbb{N}$,$\gamma, \rho \in]0, +\infty[$, and $\alpha, \delta, \delta_+ \in]0, 1[$
Initialization : $k_{-1} = 1$, $n = 0$, and $x_0 \in \mathbb{R}^d$
for $n < N$ **do**
  $k = k_{n-1}$
  $\gamma_n(x_n) = \gamma \alpha^k$
  $x = x_n - \gamma_n(x_n) \nabla_x L(x_n, y_n)$
  $y = \arg\min_{z \in \mathcal{Y}} L(x, z)$
  if $L(x, y) < L(x_n, y_n) - \delta^+ \gamma_n(x_n) \|\nabla f(x_n)\|^2$ **then**
    $k = k - 1$
  while $L(x, y) > L(x_n, y_n) - \delta \gamma_n(x_n) \|\nabla_x L(x_n, y_n)\|^2$ **do**
    $k = k + 1$
    $\gamma_n(x_n) = \gamma \alpha^k$
    $x = x_n - \gamma_n(x_n) \nabla_x L(x_n, y_n)$
    $y = \arg\min_{z \in \mathcal{X}} L(x, z)$
  $k_n = k, \quad x_{n+1} = x, \quad y_{n+1} = y.$

5.3.2 Wasserstein GAN

As explained in Section 5.2, this problem does not formally match our setting. In particular, the argmax cannot be computed fast, so we use a gradient ascent to provide an approximation expressed by using the sign $\approx$. We also provide a constant step size method (Algorithm 7) to benchmark our algorithm.

Besides, since the max is not easily accessible, we modify the while-loop by using y_n instead of the exact argmax to validate the sufficient decrease. This approach gives Algorithm 8. This is an heuristic approach which we consider here in order to provide experimental results on a real Wasserstein GAN problem. Indeed the maximization oracle is replaced by gradient ascent steps and the backtracking loop does not correspond to the one which was studied in Theorem 15. Besides, contrary to Sinkhorn GANs, nothing ensures that the inner maximum is uniquely attained in (18). This provides an initial motivation regarding further research on the topic.

Algorithm 7: Heuristic gradient method for min-max with constant step size

Input: $\gamma \in]0, +\infty[$
Initialization: $x_0 \in \mathbb{R}^d$
for $n = 0, 1, \ldots$ **do**
  $y_n \approx \operatorname{argmax}_{y \in \mathcal{Y}} L(x_n, y)$
  $x_{n+1} = x_n - \gamma \nabla_x L(x_n, y_n)$

Let us emphasize that the training of Wasserstein GANs is a challenging task in practice and is also difficult to analyse from a theoretical perspective as the models considered in most application are very large and complex.

Algorithm 8: Heuristic Hölder Backtrack for min-max

Input: $\gamma, \rho \in]0, +\infty[$ and $\delta, \alpha \in]0, 1[$
Initialization: $x_0 \in \mathbb{R}^d$
for $n = 0, 1, \ldots$ **do**
  $y_n \approx \operatorname{argmax}_{y \in \mathcal{Y}} L(x_n, y)$
  $k = 0$
  $\gamma_n(x_n) = \gamma$
  while $L(x_n - \gamma_n(x_n)\nabla_x L(x_n, y_n), y_n) > L(x_n, y_n) - \delta\gamma_n(x_n)\|\nabla_x L(x_n, y_n)\|^2$ **do**
    $k = k + 1$
    $\gamma_n(x_n) = \gamma\alpha^k \min\{1, \|\nabla_x L(x_n, y_n)\|^{k\rho}\}$
  $k_n = k$
  $x_{n+1} = x_n - \gamma_n(x_n)\nabla_x L(x_n, y_n)$

References

1 Pierre-Antoine Absil, Robert Mahony, and Benjamin Andrews. Convergence of the iterates of descent methods for analytic cost functions. *SIAM J. Optim.*, 16(2):531–547, 2005.
2 Cem Anil, James Lucas, and Roger Grosse. Sorting Out Lipschitz Function Approximation. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 291–301. PMLR, 2019.
3 Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein Generative Adversarial Networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 2017.
4 Hédy Attouch, Jérôme Bolte, Patrick Redont, and Antoine Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-łojasiewicz inequality. *Math. Oper. Res.*, 35(2):438–457, 2010.
5 Hedy Attouch, Jérôme Bolte, and Benar Fux Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods. *Math. Program.*, 137(1-2):91–129, 2013.
6 Guillaume O. Berger, Pierre-Antoine Absil, Raphaël M. Jungers, and Yurii Nesterov. On the Quality of First-Order Approximation of Functions with Hölder Continuous Gradient. *J. Optim. Theory Appl.*, 185(1):17–33, 2020.
7 Dimitri P. Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Academic Press Inc., 2014.
8 Jacek Bochnak, Michel Coste, and Marie-Françoise Roy. *Géométrie algébrique réelle*, volume 12. Springer, 1987.
9 Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal Alternating Linearized Minimization for Nonconvex and Nonsmooth Problems. *Math. Program.*, 146(1-2):459–494, 2014.
10 Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Nonconvex Lagrangian-based optimization: monitoring schemes and global convergence. *Math. Oper. Res.*, 43(4):1210–1232, 2018.
11 Stephen P. Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
12 Camille Castera, Jérôme Bolte, Cédric Févotte, and Edouard Pauwels. An Inertial Newton Algorithm for Deep Learning. `https://arxiv.org/abs/1905.12278`, 2019.
13 Patrick L. Combettes and Heinz H. Bauschke. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2nd ed. edition, 2017.
14 Michel Coste. An Introduction to Semialgebraic Geometry. *RAAG Notes*, 1999.
15 Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, pages 2292–2300. Curran Associates Inc., 2013.

16 Aude Genevay, Gabriel Peyré, and Marco Cuturi. Gan and VAE from an Optimal Transport Point of View. `https://arxiv.org/abs/1706.01807`, 2017.

17 Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning Generative Models with Sinkhorn Divergences. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1608–1617. PMLR, 2018.

18 Gauthier Gidel, Hugo Berard, Gaëtan Vignoud, Pascal Vincent, and Simon Lacoste-Julien. A Variational Inequality Perspective on Generative Adversarial Networks. In *International Conference on Learning Representations*, 2019.

19 Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pages 2672–2680. Curran Associates, Inc., 2014.

20 Geovani N. Grapiglia and Yurii Nesterov. Tensor Methods for Minimizing Functions with Hölder Continuous Higher-Order Derivatives. `https://arxiv.org/abs/1904.12559`, 2019.

21 Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved Training of Wasserstein GANs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5769–5779. Curran Associates Inc., 2017.

22 Yu-Guan Hsieh, Franck Iutzeler, Jérôme Malick, and Panayotis Mertikopoulos. On the convergence of single-call stochastic extra-gradient methods. In *Proceedings of the 32th International Conference on Neural Information Processing Systems*, pages 6936–6946, 2019.

23 G. M. Korpelevich. An extragradient method for finding saddle points and for other problems. *Ehkon. Mat. Metody*, 12(4):747–756, 1976.

24 Krzysztof Kurdyka. On gradients of functions definable in o-minimal structures. *Ann. Inst. Fourier*, 48(3):769–783, 1998.

25 Rida Laraki, Jérôme Renault, and Sylvain Sorin. *Mathematical foundations of game theory*. Springer, 2019.

26 Tianyi Lin, Chi Jin, and Michael I. Jordan. On Gradient Descent Ascent for Nonconvex-Concave Minimax Problems. `https://arxiv.org/abs/1906.00331`, 2020.

27 Panayotis Mertikopoulos, Bruno Lecouat, Houssam Zenati, Chuan-Sheng Foo, Vijay Chandrasekhar, and Georgios Piliouras. Optimistic mirror descent in saddle-point problems: Going the extra(-gradient) mile. In *International Conference on Learning Representations*, 2019.

28 Arkadi Nemirovski. Prox-Method with Rate of Convergence $O(1/t)$) for Variational Inequalities with Lipschitz Continuous Monotone Operators and Smooth Convex-Concave Saddle Point Problems. *SIAM J. Optim.*, 15(1):229–251, 2004.

29 Yu Nesterov. Universal Gradient Methods for Convex Optimization Problems. *Math. Program.*, 152(1-2):381–404, 2015.

30 Maher Nouiehed, Maziar Sanjabi, Tianjian Huang, Jason D. Lee, and Meisam Razaviyayn. Solving a Class of Non-Convex Min-Max Games Using Iterative First Order Methods. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Proceedings of the 32th International Conference on Neural Information Processing Systems*, pages 14905–14916. Curran Associates, Inc., 2019.

31 Maher Nouiehed, Maziar Sanjabi, Tianjian Huang, Jason D. Lee, and Meisam Razaviyayn. Solving a Class of Non-Convex Min-Max Games Using Iterative First Order Methods. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Proceedings of the 32th International Conference on Neural Information Processing Systems*, pages 14934–14942. Curran Associates, Inc., 2019.

32 R. Tyrrell Rockafellar. Proximal subgradients, marginal values, and augmented Lagrangians in nonconvex optimization. *Math. Oper. Res.*, 6(3):424–436, 1981.

33 R. Tyrrell Rockafellar and Roger J.-B. Wets. *Variational Analysis*. Number 317 in Grundlehren der Mathematischen Wissenschaften. Springer, 1998.

34 Shoham Sabach and Marc Teboulle. Lagrangian Methods for Composite Optimization. In *Processing, analyzing and learning of images, shapes, and forms. Part 2*, volume 20 of *Handbook of Numerical Analysis*, pages 401–436. Elsevier, 2019.

35 Richard Sinkhorn. A Relationship Between Arbitrary Positive Matrices and Doubly Stochastic Matrices. *Ann. Math. Stat.*, 35(2):876–879, 1964.

36 John Von Neumann. Zur theorie der gesellschaftsspiele. *Math. Ann.*, 100(1):295–320, 1928.

37 John Von Neumann. On Rings of Operators. Reduction Theory. *Ann. Math.*, 50(2):401–485, 1949.

38 Jingkang Wang, Tianyun Zhang, Sijia Liu, Pin-Yu Chen, Jiacen Xu, Makan Fardad, and Bo Li. Towards A Unified Min-Max Framework for Adversarial Exploration and Robustness. `https://arxiv.org/abs/1906.03563`, 2019.

39 Maryam Yashtini. On the Global Convergence Rate of the Gradient Descent Method for Functions with Hölder Continuous Gradients. *Optim. Lett.*, 10(6):1361–1370, 2016.