An implicit gradient-descent procedure for minimax problems

Montacer Essid[1†], Esteban G. Tabak[1†] and Giulio Trigila[2*†]

[1]Courant Institute of Mathematical Sciences, 251 Mercer Street, New York City, 10012, NY, USA.
[2*]Baruch College, One Bernard Baruch Way, New York City, 10010, NY, USA.

*Corresponding author(s). E-mail(s): giulio.trigila@baruch.cuny.edu;
Contributing authors: essid@cims.nyu.edu; tabak@cims.nyu.edu;
†These authors contributed equally to this work.

Abstract

A game theory inspired methodology is proposed for finding a function's saddle points. While explicit descent methods are known to have severe convergence issues, implicit methods are natural in an adversarial setting, as they take the other player's optimal strategy into account. The implicit scheme proposed has an adaptive learning rate that makes it transition to Newton's method in the neighborhood of saddle points. Convergence is shown through local analysis and through numerical examples in optimal transport and linear programming. An ad-hoc quasi-Newton method is developed for high dimensional problems, for which the inversion of the Hessian of the objective function may entail a high computational cost.

Keywords: Saddle point optimization, adversarial optimization, game theory, optimal transport.

1 Introduction

Saddle point problems occur in a wide variety of applications ranging from economics ([1, 2]), where competing behaviours between two players are modeled by means of zero-sum games, mechanics and computational fluid dynamics

[3, 4], where the numerical solution of partial differential equations can be approached studing saddle point problems, to constrained optimization [5–7], where enforcing a constraint while optimizing a given objective function is achieved through a min-maximization of the Lagrangian function.

More recently, saddle point problems have played a crucial role in machine learning (ML) and statistics [8–12], particularly in the context of generative adversarial neural networks (GANs) [13] and optimal transport [14].

The success of GANs is based on their ability to indirectly train a model represented by a neural network through two players, a generator of data mimicking the training set and a discriminator that seeks differences between the data generated and the training set. This enables fundamental tasks such as regression, density and conditional density estimation, with applications in the natural sciences ranging from chemistry [15, 16] to genomics [17] and neuroscience [18], as well as more established ML tasks such as imaging analysis [19]. Optimal transport [20–23], the corresponding barycenter problem and its connection to normalizing flows ([24–26]) provide another, emerging toolbox for machine learning where the numerical solution of saddle point problems plays a critical role.

The general structure of a saddle point problem is formulated in terms of the mini-maximization of a Lagrangian function:

$$\min_x \max_y L(x, y) \tag{1}$$

where typically $x \in R^{n_x}$ and $y \in R^{n_y}$ or subsets thereof. The following is a list of examples directly related to the discussions below:

Example 1: equality-constrained minimization

$$\min_x f(x) \quad \text{subject to} \quad g(x) = 0.$$

Introducing Lagrange multipliers y yields

$$\min_x \max_y L(x, y) = f(x) - y^t g(x).$$

Often some components of x and y are required to be non-negative:

Example 2: inequality-constrained minimization

$$\min_x f(x) \quad \text{subject to} \quad g(x) \geq 0,$$

where introducing Lagrange multipliers y yields

$$\min_x \max_{y \geq 0} L(x, y) = f(x) - y^t g(x).$$

When there are both equality and inequality constraints, only the Lagrange multipliers y_j attached to the inequalities are required to be non-negative.

Example 3: two-player zero-sum games

$$\min_x \max_y y^t Ax, \quad \text{with } x, y \geq 0, \quad \sum_i x_i = \sum_j y_j = 1.$$

Introducing Lagrange multipliers λ and μ for the equality constraints, yields

$$\min_{x \geq 0, \mu} \max_{y \geq 0, \lambda} \; L = y^t \; A \; x - \lambda \left(\sum_i x_i - 1 \right) - \mu \left(\sum_j y_j - 1 \right).$$

A more recent development formulates problems of interest as nonlinear adversarial games. Examples include generative adversarial networks [13], as well as the following [22]:

Example 4: adaptive optimal transport

$$\min_\alpha \max_\beta \left[\sum_i w_i^x \; g \left(\nabla \phi \left(x_i, \alpha \right), \beta \right) - \sum_j w_j^y \; e^{g(y_j, \beta)} \right],$$

where α parameterizes a curl-free map $T = \nabla \phi(x)$ that pushes forward the distribution underlying the samples x_i to the one underlying the y_j, and β parameterizes a test function $g(y)$ that enforces the push-forward constraints.

This article proposes a game-theory inspired methodology for the numerical solution of minimax problems, *implicit twisted-gradient descent*: "twisted" because one player descends the gradient while the other ascends it, and "implicit" because the two players simultaneously descend (in x) and ascend (in y) the Lagrangian L in an anticipatory manner, i.e. following the gradient of L estimated at the values of (x, y) resulting from the current step. For small learning rates η, each step of this procedure converges to regular (twisted) gradient descent, while for large η it converges to a Newton step.

Previous work on minimax solvers generally requires strict convexity-concavity of the objective function [3, 27, 28]. There are analogies between the work presented here and the proposal in [29], which implements a twisted gradient descent by an ad hoc modification of the mirror descent method. This modification is based on predictable sequences in which at each time step a guess on the future direction of the gradient is made. In the methodology proposed here, the anticipation of the next gradient uses the Hessian, making it possible to leverage the extensive optimization literature on Newton's method. An example in this direction is the development of a quasi-Newton-like method presented in Section 5. Another method exploiting the curvature of the objective function is presented in [30] where, similarly to what is done here, the curvature is used to escape regions of undesired stability of the twisted gradient method, using only the smallest and the largest eigenvalue of the Hessian of L. As in [30], we use a definition of local saddle points that we further refine by introducing at the beginning of Section 3 the definition of strict local saddle points, allowing us to establish the local convergence of our algorithm. The work in [31] also uses the Hessian of L to develop an algorithm different from ours that can be proved to converge to a local saddle point. Since the adoption of quasi Newton methods is not as straightforward as in our setting, the authors use a two-scale algorithm to avoid inverting the Hessian. The Hessian is also used in [32] to develop an algorithm converging to non-convex in x –but concave in y– saddle points. The algorithm presented here appeared for the first time in [33]. In this article we complete the algorithm with a discussion on its local convergence and we further refine the criterion for the choice of the learning rate η.

The plan of the article is as follows: after this introduction, section 2 introduces the basic step of the procedure for a given learning rate η. Section 3 proves the procedure's local convergence. Section 4 proposes an adaptive criterion for evolving the learning rate. Section 5 develops a quasi-Newton-like methodology that bypasses the need to evaluate the Hessian of L or to invert any matrix. Section 6 extends the methodology to situation where some or all variables are required to be positive. Section 7 shows examples of numerical results. Finally, section 8 includes some concluding remarks.

2 Implicit twisted gradient descent

We consider first the case without positivity constraints:

$$\min_x \max_y L(x,y), \quad x \in R^{n_x}, \quad y \in R^{n_y}. \tag{2}$$

An explicit, twisted gradient descent step is given by

$$\begin{aligned} x^{n+1} &= x^n - \eta L_x|_{x^n,y^n} \\ y^{n+1} &= y^n + \eta L_y|_{x^n,y^n}, \end{aligned} \tag{3}$$

where $L_x = \nabla_x L$, $L_y = \nabla_y L$, and $\eta > 0$ is the learning rate: the players with strategy x and y seek to decrease and increase L respectively, and do so following the direction of their components of the gradient of L. For compactness, we introduce the following notation:

$$z = \begin{pmatrix} x \\ y \end{pmatrix}, \quad G = \begin{pmatrix} L_x \\ L_y \end{pmatrix}, \quad J = \begin{pmatrix} I_x & 0 \\ 0 & -I_y \end{pmatrix}, \quad L^n = L(x^n, y^n), \tag{4}$$

where I_x and I_y are identity matrices of size n_x and n_y respectively. Then the descent step reads

$$z^{n+1} = z^n - \eta \ J \ G^n. \tag{5}$$

Yet such a procedure may fail to converge [34]. Consider the simple example with $L = xy$, which has the unique mini-maximizer $x = y = 0$. Here twisted gradient descent yields

$$x^{n+1} = x^n - \eta y^n$$

$$y^{n+1} = y^n + \eta x^n,$$

or

$$z^{n+1} = \begin{pmatrix} 1 & -\eta \\ \eta & 1 \end{pmatrix} z^n,$$

which diverges, since the matrix eigenvalues $\lambda^{\pm} = 1 \pm i\eta$ have absolute value greater than 1. Even in the limit of infinitesimally small values of η, the solution moves in circles around the origin, following the system of ODEs

$$\begin{aligned} \dot{x} &= -y \\ \dot{y} &= x. \end{aligned}$$

One could argue that, from a game-theory perspective, each player would not merely move following their own local gradient, but would try to anticipate how the other player will move. This suggests a form of implicit twisted gradient descent:

$$z^{n+1} = z^n - \eta \; J \; G^{n+1}. \tag{6}$$

Applying (6) to the example above yields

$$\begin{pmatrix} 1 & \eta \\ -\eta & 1 \end{pmatrix} z^{n+1} = z^n,$$

with unconditional convergence to $z = 0$, at a convergence rate that grows unboundedly with the learning rate η.

Yet in general (6) cannot be solved in closed form for z^{n+1}. Instead, we may approximate G^{n+1} using

$$G^{n+1} \approx G^n + H^n \left(z^{n+1} - z^n \right), \tag{7}$$

where H is the Hessian

$$H = \begin{pmatrix} L_{xx} & L_{xy} \\ L_{yx} & L_{yy} \end{pmatrix}, \tag{8}$$

$$(L_{xx})_i^j = \frac{\partial^2 L}{\partial x_i \partial x_j}, \quad (L_{yy})_i^j = \frac{\partial^2 L}{\partial y_i \partial y_j}, \quad (L_{xy})_i^j = \frac{\partial^2 L}{\partial x_i \partial y_j}, \quad L_{yx} = L_{xy}^t.$$

Under this approximation, the scheme in (6) yields

$$z^{n+1} = z^n - \eta \; J \; \left(G^n + H^n \left(z^{n+1} - z^n \right) \right),$$

which reduces to

$$z^{n+1} = z^n - \eta \left(J + \eta H^n \right)^{-1} G^n. \tag{9}$$

This is the basic updating step of the proposed algorithm.

Notice that, as $\eta \to \infty$, the update in (9) converges to the Newton step

$$z^{n+1} = z^n - \left(H^n \right)^{-1} G^n$$

while, for small values of η, it approximates the explicit twisted gradient descent step in (5).

This proposal leaves us with some tasks:

1. Prove convergence of the algorithm,
2. develop a scheme for updating the learning rate η so as to accelerate convergence to saddle points and impede convergence to non-saddle critical points of L,
3. develop a way to avoid inverting possibly large matrices and, if possible, avoid computing the Hessian altogether,
4. extend the procedure to situations where some or all variables have positivity constraints, and
5. show numerical examples of the algorithm at work.

These tasks are addressed in the following sections.

3 Local convergence

We begin this section with the definition of strict local saddle point:

Definition 1 We define a point (x^*, y^*) to be a strict local mini-maximizer of $L(x, y)$ if, for any sufficiently small neighborhoods U_x of x^* and U_y of y^*,

$$\forall x \in U_x\ \forall y \in U_y,\ L\left(x^*, y\right) \leq L\left(x^*, y^*\right) \leq L\left(x, y^*\right) \tag{10}$$

and

$$\forall x \in U_x,\ x \neq x^* \Rightarrow \exists y \in U_y\ /\ L(x, y) > L\left(x^*, y^*\right), \tag{11}$$

$$\forall y \in U_y,\ y \neq y^* \Rightarrow \exists x \in U_x\ /\ L(x, y) < L\left(x^*, y^*\right). \tag{12}$$

Conditions (11) and (12) exclude areas where L is flat in x or y, which could yield a continuum of [non-strict] local minimax points.

For smooth Lagrangians L, if at a point (x^*, y^*), $G = 0$ and the two block diagonal elements of the Hessian, L_{xx} and L_{yy}, are respectively positive and definite negative, then (x^*, y^*) is a strict local mini-maximizer of $L(x, y)$. However, we are interested in situations more general than this, for instance when L depends linearly on some of the variables. This case is not merely of academic interest: it arises frequently when the variables are Lagrange multipliers associated to constraints in a minimization problem and in zero-sum games. Consider for instance the simplest nontrivial Lagrangian, $L = xy$, whose Hessian vanishes and which nonetheless admits the unique global mini-maximizer $x = y = 0$. Thus we need to consider situations where some eigenvalues of L_{xx} or L_{yy} may vanish, so the off-block diagonal terms L_{xy} and L_{yx} in the Hessian become key to the existence of a saddle point.

If L is smooth at a point (x^*, y^*), then locally

$$L(x, y) - L = L_x \Delta x + L_y \Delta y + \frac{1}{2} \Delta x^t L_{xx} \Delta x + \Delta x^t L_{xy} \Delta y + \frac{1}{2} \Delta y^t L_{yy} \Delta y$$

$$+ O\left(\|\Delta z\|^3\right),$$

where $\Delta z = (\Delta x, \Delta y) = (x - x^*, y - y^*)$, and L, L_x, L_{xx}, etc. stand for their values at (x^*, y^*). It follows that the following are sufficient conditions for the point (x^*, y^*) to be a strict local mini-maximizer.

Theorem 1 *The following conditions guarantee that $z^* = (x^*, y^*)$ is a strict local mini-maximizer of a smooth function $L(x, y)$:*

1. *The conditions in (10) are satisfied. This implies in particular that L_x and L_y vanish and that L_{xx} is positive semi-definite and L_{yy} is negative semi-definite.*
2. *The null spaces of L_{xx} and L_{yx} are orthogonal to each other, as are the null spaces of L_{yy} and L_{xy}.*

Let N_{xx}, N_{yx}, $N_{xx}^{\perp}$ and $N_{yx}^{\perp}$ denote the null spaces of L_{xx} and L_{yx} and their orthogonal complements, and similarly for the null spaces of L_{yy} and L_{xy}. Notice that the conditions in *2* include as particular cases the following situations (with a simple example attached to each):

- L_{xx} is positive definite and L_{yy} negative definite, as then both N_{xx} and N_{yy} reduce to the zero vector (e.g. $L = x^2 - y^2$),
- L_{xx} is positive definite and L_{xy} is injective, making N_{xx} and N_{xy} trivial (e.g. $L = x_1^2 + x_2^2 + (x_1 + 2x_2)y$),
- L_{yy} is negative definite and L_{yx} is injective (e.g. $L = y_1^2 + (y_2 - 1)^2 + (y_1 - y_2)x$),
- L_{xy} and L_{yx} are both injective, for which X and Y would need to have the same dimension (e.g. $L = xy$).

In fact, weaker conditions are sufficient, requiring not the orthogonality among subspaces, but only that the intersections of N_{xx} and N_{yx} and of N_{yy} and N_{xy} consist only of the zero vectors in X and Y respectively. For clarity though, we only write here the proof for the theorem as stated above, relegating the proof of the stronger result to appendix A.

Proof It is clear that the conditions in (10) require L_x and L_y to vanish, and L_{xx} and L_{yy} to be at least positive and negative semi-definite respectively. For instance, setting $\Delta y = 0$, we have

$$L(x, y^*) = L + L_x \Delta x + \frac{1}{2} \Delta x^t L_{xx} \Delta x + O\left(\|\Delta x\|^3\right),$$

so unless $L_x = 0$ and L_{xx} is positive semidefinite, there will be values of $x \in U_x$ with $L(x, y^*) < L(x^*, y^*)$, contradicting (10).

Because X and Y are finite-dimensional, there exist positive constants s_{xx}, s_{yx}, s_{yy} and non-negative constants S_{xx}, S_{yx}, and S_{yy} such that

$$a \in N_{xx}^{\perp} \Rightarrow \|L_{xx} a\| \geq s_{xx} \|a\|, \quad b \in N_{yx}^{\perp} \Rightarrow \|L_{yx} b\| \geq s_{yx} \|b\|,$$

$$c \in N_{yy}^{\perp} \Rightarrow \|L_{yy} c\| \geq s_{yy} \|c\|, \quad d \in N_{xy}^{\perp} \Rightarrow \|L_{xy} d\| \geq s_{yx} \|d\|,$$

$$\|L_{yx}v\| \leq S_{yx}\|v\|, \quad v^t L_{xx} v \leq S_{xx}\|v\|^2,$$
$$\|L_{xy}w\| \leq S_{yx}\|w\|, \quad -w^t L_{yy} w \leq S_{yy}\|w\|^2,$$

where the s and S are the smallest (non-zero) and largest singular values of the corresponding matrices.

Consider now a point $x \neq x^*$. We can write (uniquely)

$$\Delta x = x - x^* = x_1 + x_2 = x_3 + x_4,$$

where

$$x_1 \in N_{xx}, \quad x_2 \in N_{xx}^{\perp}, \quad x_3 \in N_{yx}, \quad x_4 \in N_{yx}^{\perp}.$$

Moreover, the fact that N_{xx} and N_{yx} are orthogonal implies that

$$\max\left(\|x_2\|^2, \|x_4\|^2\right) \geq \frac{1}{2}\|\Delta x\|^2, \tag{13}$$

as follows from the following argument:

$$\|x_2\|^2 \leq \frac{1}{2}\|\Delta x\|^2 \Rightarrow \|x_1\|^2 \geq \frac{1}{2}\|\Delta x\|^2,$$

since $\|x_1\|^2 + \|x_2\|^2 = \|\Delta x\|^2$. But

$$\|x_1\|^2 \geq \frac{1}{2}\|\Delta x\|^2 \Rightarrow \|x_4\|^2 \geq \frac{1}{2}\|\Delta x\|^2,$$

since $N_{xx} \subseteq N_{yx}^{\perp}$ from the orthogonality of N_{xx} and N_{yx}, thus proving (13).

This inequality allows us to consider two scenarios:

1. If $\|x_2\|^2 \geq \frac{1}{2}\|\Delta x\|^2$, we can set $\Delta y = 0$, and obtain

$$\begin{aligned} L(x,y) &= L(x^*,y^*) + \frac{1}{2}\Delta x^t L_{xx}\Delta x + \ O\left(\|\Delta x\|^3\right) \\ &= L(x^*,y^*) + \frac{1}{2}x_2^t L_{xx} x_2 + \ O\left(\|\Delta x\|^3\right) \\ &\geq L(x^*,y^*) + \frac{s_{xx}}{8}\|\Delta x\|^2 + \ O\left(\|\Delta x\|^3\right) \\ &> L(x^*,y^*) \end{aligned}$$

for small enough $\|\Delta x\|$, as required by condition (11).

2. If $\|x_2\|^2 \leq \frac{1}{2}\|\Delta x\|^2$, it follows from (13) that $\|x_4\|^2 \geq \frac{1}{2}\|\Delta x\|^2$. Then adopting

$$\Delta y = \alpha L_{yx}\Delta x = \alpha L_{yx} x_4,$$

with

$$\alpha = \min\left(\frac{s_{yx}^2}{S_{yx}^2 S_{yy}}, \alpha_{max}\right),$$

where α_{max} is small enough for $y = y^* + \Delta y$ to lie within U_y, yields

$$\begin{aligned} L(x,y) &= L\left(x^*,y^*\right) + \frac{1}{2}\Delta x^t L_{xx}\Delta x + \Delta x^t L_{xy}\Delta y + \frac{1}{2}\Delta y^t L_{yy}\Delta y + O\left(\|\Delta z\|^3\right) \\ &\geq L\left(x^*,y^*\right) + \Delta x^t L_{xy}\Delta y + \frac{1}{2}\Delta y^t L_{yy}\Delta y + O\left(\|\Delta z\|^3\right) \\ &= L\left(x^*,y^*\right) + \alpha\|L_{xy}x_4\|^2 + \frac{\alpha^2}{2}x_4^t L_{xy} L_{yy} L_{yx} x_4 + O\left(\|\Delta z\|^3\right) \end{aligned}$$

$$
\begin{aligned}
&\geq L\left(x^{*}, y^{*}\right)+\left[\alpha s_{yx}^{2}-\frac{\alpha^{2}}{2} S_{xy}^{2} S_{yy}\right]\|x_{4}\|^{2}+O\left(\|\Delta z\|^{3}\right) \\
&\geq L\left(x^{*}, y^{*}\right)+\frac{1}{2} \alpha s_{yx}^{2}\|x_{4}\|^{2}+O\left(\|\Delta z\|^{3}\right) \\
&> L\left(x^{*}, y^{*}\right)
\end{aligned}
$$

for Δx small enough, as follows from the upper bound

$$
\|\Delta z\|^{2}=\|\Delta x\|^{2}+\|\Delta y\|^{2} \leq\left(2+\alpha^{2} S_{yx}^{2}\right)\|x_{4}\|^{2}.
$$

The same argument, mutatis mutandis, proves condition (12).

□

It follows that, in a domain where the sufficient conditions on the Hessian from Theorem 1 apply, a local mini-maximizer z^* of L is uniquely characterized by the first order conditions

$$
G\left(z^{*}\right)=0.
$$

Thus if the conditions on the Hessian are satisfied in a neighborhood of the local optimal z^* that the algorithm does not leave, it is enough to show that, for fixed η, $\|G\|$ decreases to zero in order to guarantee convergence. This is proven below.

Theorem 2 *Suppose that a strict local mini-maximizer z^* of (2) satisfies the sufficient conditions of Theorem 1, with the conditions on the Hessian satisfied in a neighborhood of z^*. Then there exists a learning rate $\eta > 0$ such that, for any z sufficiently close to z^*, the dynamics in (9) converges to z^*.*

Proof There exists a neighborhood of z^* where $\|G(z)\| > 0$ for $z \neq z^*$, for otherwise there would be another strict local mini-maximizer arbitrarily close to z^*, which is a contradiction. This implies that there is a smaller neighborhood U_* of z^* where $\|G(z)\|$ grows as z moves away from z^*, i.e. the contour lines of $\|G\|$ enclose z^* and the corresponding values of $\|G\|$ vary monotonically across them. Thus, in order to prove convergence, it is enough to show that $\|G\|$ decreases uniformly at each step, i.e. that $\|G^{n+1}\|^2 \leq \alpha\|G^n\|^2$ for some $\alpha < 1$ and, moreover, there is a continuous path between z^n and z^{n+1} along which the values of $\|G(z)\|$ are never larger than $\|G^0\|$, guaranteeing that z does not leave U_*.

Expanding G^{n+1} and retaining only the terms up to linear in Δz as in (7),

$$
G^{n+1} \approx G^{n}+H^{n}\left(z^{n+1}-z^{n}\right)+O\left(\Delta z^{2}\right), \tag{14}
$$

the procedure in (9) yields

$$
\begin{aligned}
G^{n+1} &=\left(I-\eta H^{n}\left(J+\eta H^{n}\right)^{-1}\right) G^{n}+O\left(\Delta z^{2}\right) \\
&=J\left(J+\eta H^{n}\right)^{-1} G^{n}+O\left(\Delta z^{2}\right).
\end{aligned} \tag{15}
$$

It follows that

$$
\|G^{n}\|^{2}=\left\|G^{n+1}+\eta H^{n} J G^{n+1}\right\|^{2}+O\left(\|G^{n}\| \Delta z^{2}\right)
$$

$$\begin{aligned}
&= \left\|G^{n+1}\right\|^2 + 2\eta {G^{n+1}}^t H^n J G^{n+1} + \eta^2 \left\|H^n J G^{n+1}\right\|^2 + O\left(\|G^n\|\Delta z^2\right) \\
&= \left\|G^{n+1}\right\|^2 + 2\eta\left(u^t L_{xx} u - v^t L_{yy} v\right) + \\
&\quad + \eta^2 \Big[\, \|L_{xx}u - L_{xy}v\|^2 + \|L_{yy}v - L_{yx}u\|^2 \Big] + O\left(\|G^n\|\Delta z^2\right), \qquad (16)
\end{aligned}$$

where $u = L_x^{n+1}$ and $v = L_y^{n+1}$.

Using the same notation as in Theorem 1 for the largest and smallest singular values of the various blocks of the Hessian matrix, one can easily prove the additional inequalities

$$u^t L_{xx} u \geq \frac{1}{S_{xx}} \|L_{xx}u\|^2 \quad \text{and} \quad -v^t L_{yy} v \geq \frac{1}{S_{yy}} \|L_{yy}v\|^2$$

whenever S_{xx} and S_{yy} are respectively non-zero (else L_{xx} –resp. L_{yy}– vanishes). Also as in Theorem 1, we can decompose u and v uniquely in the form

$$u = u_1 + u_2 = u_3 + u4, \quad u_1 \in N_{xx}, \quad u_2 \in N_{xx}^{\perp}, \quad u_3 \in N_{yx}, \quad u_4 \in N_{yx}^{\perp},$$

$$v = v_1 + v_2 = v_3 + v4, \quad v_1 \in N_{yy}, \quad v_2 \in N_{yy}^{\perp}, \quad v_3 \in N_{xy}, \quad v_4 \in N_{xy}^{\perp},$$

with

$$\max\left(\|u_2\|^2, \|u_4\|^2\right) \geq \frac{1}{2}\|u\|^2, \quad \max\left(\|v_2\|^2, \|v_4\|^2\right) \geq \frac{1}{2}\|v\|^2. \qquad (17)$$

This allows us to rewrite (16) in the form

$$\begin{aligned}
\|G^n\|^2 = \left\|G^{n+1}\right\|^2 + 2\eta\left(u_2^t L_{xx} u_2 - v_2^t L_{yy} v_2\right) + +\eta^2 \Big[\, \|L_{xx}u_2 - L_{xy}v_4\|^2 \\
+ \|L_{yy}v_2 - L_{yx}u_4\|^2 \Big] + O\left(\|G^n\|\Delta z^2\right) \qquad (18)
\end{aligned}$$

and bound the various terms depending on the relative sizes of the norms of u, v, $u_{2,4}$, $v_{2,4}$, using the following corollaries of (18):

$$\|u_2\|^2 \geq \frac{1}{2}\|u\|^2 \Rightarrow \|G^n\|^2 - \left\|G^{n+1}\right\|^2 \geq \eta s_{xx} \|u\|^2, \qquad (19)$$

$$\|v_2\|^2 \geq \frac{1}{2}\|v\|^2 \Rightarrow \|G^n\|^2 - \left\|G^{n+1}\right\|^2 \geq \eta s_{yy} \|v\|^2, \qquad (20)$$

$$\|u_4\| \geq \max\left(\frac{2S_{yy}}{s_{yx}}\|v_2\|, \frac{\|u\|^2}{2}\right) \Rightarrow \|G^n\|^2 - \left\|G^{n+1}\right\|^2 \geq \frac{1}{8}\eta^2 s_{yx}^2 \|u\|^2, \qquad (21)$$

$$\|v_4\| \geq \max\left(\frac{2S_{xx}}{s_{yx}}\|u_2\|, \frac{\|v\|^2}{2}\right) \Rightarrow \|G^n\|^2 - \left\|G^{n+1}\right\|^2 \geq \frac{1}{8}\eta^2 s_{yx}^2 \|v\|^2, \qquad (22)$$

where the right-hand sides of all four inequalities are up to corrections of order $O\left(\|G^n\|\Delta z^2\right)$ or, equivalently, $O\left(\eta^2\|G^n\|^3\right)$, since $\Delta z \approx \eta G^n$.

Since our goal is to bound $\Delta = \|G^n\|^2 - \left\|G^{n+1}\right\|^2$ below by a constant times $\left\|G^{n+1}\right\|^2 = \|u\|^2 + \|v\|^2$, we will seek lower bounds for Δ proportional to $\|u\|^2$ and to $\|v\|^2$, and then combine these into a bound proportional to $\left\|G^{n+1}\right\|^2$. Let us consider lower bounds proportional to $\|u\|^2$ first, as the ones proportional to $\|v\|^2$ derive from an identical argument.

If $\|u_2\|^2 \geq \frac{1}{2}\|u\|^2$, then it follows from (19) that

$$\Delta \geq \eta s_{xx}\|u\|^2 + O\left(\eta^2\|G^n\|^3\right).$$

Otherwise, it follows from (17) that $\|u_4\|^2 \geq \frac{1}{2}\|u\|^2$. If in addition $\|u_4\| \geq \frac{2S_{yy}}{s_{yx}}\|v_2\|$, then it follows from (21) that

$$\Delta \geq \frac{1}{8}\eta^2 s_{yx}^2 \|u\|^2 + O\left(\eta^2 \|G^n\|^3\right).$$

Otherwise, i.e. if $\|u_4\| < \frac{2S_{yy}}{s_{yx}}\|v_2\|$, it follows from (18) that

$$\Delta \geq -2\eta v_2^t L_{yy} v_2 \geq \frac{2\eta}{S_{yy}}\|L_{yy}v_2\|^2 \geq \frac{2s_{yy}^2\eta}{S_{yy}}\|v_2\|^2 \geq \eta\left(\frac{s_{yy}}{S_{yy}}\right)^2\|u_4\|^2 \geq \frac{\eta}{2}\left(\frac{s_{yy}}{S_{yy}}\right)^2\|u\|^2$$
$$+O\left(\eta^2\|G^n\|^3\right).$$

So, in all cases, if η is bounded below by a positive constant, we have established a bound of the form

$$\Delta \geq a\|u\|^2 + O\left(\eta^2\|G^n\|^3\right), \quad a > 0,$$

as required. An identical argument provides a bound

$$\Delta \geq b\|v\|^2 + O\left(\eta^2\|G^n\|^3\right), \quad b > 0,$$

and therefore

$$\|G^n\|^2 - \left\|G^{n+1}\right\|^2 \geq \frac{1}{2}\min(a,b)\left\|G^{n+1}\right\|^2 + O\left(\eta^2\|G^n\|^3\right),$$

which guarantees convergence to $G = 0$ if the learning η is bounded below. Notice that, for smaller values of η, we still have $\|G^{n+1}\| \leq \|G^n\|$. Thus the path in z-space corresponding to learning rates between 0 and η never leaves U_*.

□

Theorem 2 establishes convergence of the algorithm in the neighborhood of a strict local mini-maximizer of L. In addition, one would like the algorithm to converge only to such local mini-maximizers. To see that this is not at all guaranteed, notice that, as the learning rate η grows unboundedly, the algorithm becomes Newton's, which converges locally to zeros of G irrespective of whether these correspond to saddle points, maxima or minima of L. Thus, while near a local mini-maximizer of L one should adopt a very large value of η in order to enjoy the fast-convergence associated to Newton, one should be careful not to make η too large far from local mini-maximizers, as this may result in convergence to a critical point of the wrong type.

For example, the Lagrangian

$$L(x,y) = \frac{1}{2}\left(x^2 + y^2\right)$$

has no finite mini-maximizer, yet (9) yields

$$\begin{pmatrix} x^{n+1} \\ y^{n+1} \end{pmatrix} = \begin{pmatrix} \frac{1}{1+\eta} & 0 \\ 0 & \frac{1}{1-\eta} \end{pmatrix} \begin{pmatrix} x^n \\ y^n \end{pmatrix}, \tag{23}$$

which, for $\eta > 2$, converges to $(x, y) = (0, 0)$. We develop in section 4 a strategy for bounding η so that such spurious convergence cannot take place.

We close this section with a remark regarding the non-asymptotic convergence of the implicit scheme in (9). Since the scheme reduces to explicit descent/ascent for small learning rates, one should expect the algorithm to inherit the non-asymptotic linear convergence of the explicit scheme in (3), proved in [35, 36]. Moreover, since the constants a and b in the lower bound on $\|G^{n+1}\|^2 - \|G^n\|^2$ established in Theorem 2 scale as η^2, which increases unboundedly near convergence, when the algorithm approaches Newton's, we could expect faster than linear convergence. Yet a rigorous analysis of the algorithm's non-asymptotic convergence is beyond the scope of this article.

4 Determination of the learning rate

In order to turn the implicit gradient descent (9) into an algorithm, one needs a mechanism to decide at each step which learning rate η to use.

From the arguments above, once close enough to the optimum, one should increase η as much as possible so as to accelerate convergence, with $\eta = \infty$ yielding Newton's method. Yet Newton's method is blind to whether one is minimizing or maximizing the objective function. In our minimax context, it could converge to points where $G = 0$ that are not minima over the x and maxima over the y. Therefore, a mechanism to control the value of η is required. Unlike in pure minimization scenarios, we cannot use the decrease of the objective function as an acceptance test. However, a simple extension applies: one can require every step to satisfy the conditions

$$L\left(x^{n+1}, y^n\right) \leq L\left(x^{n+1}, y^{n+1}\right) \leq L\left(x^n, y^{n+1}\right). \tag{24}$$

These agree with the anticipatory game idea underlying the method: given y^{n+1}, the player with strategy x should make sure to decrease L, and given x^{n+1}, the player with strategy y should make sure to increase L. Thus a step not satisfying the conditions in (24) should be rejected.

To see the effect of these constraints on the convergence of the algorithm, consider three simple prototypical examples where a closed expression for them can be derived:

1. $L = xy$,
2. $L = \frac{x^2 - y^2}{2}$,
3. $L = \frac{x^2 + y^2}{2}$.

The first represents a saddle point not satisfying the regular convexity conditions $L_{xx} > 0$, $L_{yy} < 0$, yet having a global solution ($x = y = 0$), the second does satisfy these conditions globally, and the third has no solution, so we would like y to blow up: with no local minimax solution, the algorithm should explore other areas of (x, y)-space.

1. $L = xy$ has

$$G = \begin{pmatrix} y \\ x \end{pmatrix}, \quad H = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad J = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

so the update rule (9) yields

$$\begin{pmatrix} x^{n+1} \\ y^{n+1} \end{pmatrix} = \begin{pmatrix} x^n \\ y^n \end{pmatrix} - \eta \left(J + \eta H\right)^{-1} \begin{pmatrix} y^n \\ x^n \end{pmatrix} = \frac{1}{1+\eta^2} \begin{pmatrix} x^n - \eta y^n \\ y^n + \eta x^n \end{pmatrix}.$$

Notice that here the larger η the better, as increasing η brings us closer to the solution $(0,0)$. Now considering the conditions in (24), we have

$$L(x^{n+1}, y^n) = \frac{(x^n - \eta y^n) y^n}{1+\eta^2}, \quad L(x^n, y^{n+1}) = \frac{(y^n + \eta x^n) x^n}{1+\eta^2},$$

$$L(x^{n+1}, y^{n+1}) = \frac{(y^n + \eta x^n)(x^n - \eta y^n)}{(1+\eta^2)^2},$$

so

$$L(x^{n+1}, y^{n+1}) - L(x^{n+1}, y^n) = \frac{\eta}{(1+\eta^2)^2} (x^n - \eta y^n)^2$$

and

$$L(x^n, y^{n+1}) - L(x^{n+1}, y^{n+1}) = \frac{\eta}{(1+\eta^2)^2} (y^n + \eta x^n)^2,$$

both non-negative for all positive η, hence imposing no restrictions on the learning rate. This is in line with the fact that, in this case, the solution of the problem can be reached in just one step by adopting $\eta = \infty$.

2. $L = \frac{1}{2}\left(x^2 - y^2\right), \quad G = \begin{pmatrix} x \\ -y \end{pmatrix}, \quad H = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$

$$\begin{pmatrix} x^{n+1} \\ y^{n+1} \end{pmatrix} = \begin{pmatrix} x^n \\ y^n \end{pmatrix} - \eta \left(J + \eta H\right)^{-1} \begin{pmatrix} x^n \\ -y^n \end{pmatrix} = \frac{1}{1+\eta} \begin{pmatrix} x^n \\ y^n \end{pmatrix}.$$

Again, the larger η the better. On the other hand, we have

$$L(x^{n+1}, y^n) = \frac{1}{2}\left(\frac{(x^n)^2}{(1+\eta)^2} - (y^n)^2\right), \quad L(x^n, y^{n+1}) = \frac{1}{2}\left((x^n)^2 - \frac{(y^n)^2}{(1+\eta)^2}\right),$$

$$L(x^{n+1}, y^{n+1}) = \frac{1}{2}\frac{1}{(1+\eta)^2}\left((x^n)^2 - (y^n)^2\right).$$

Then both

$$L(x^{n+1}, y^{n+1}) - L(x^{n+1}, y^n) = \frac{1}{2}\frac{(1+\eta)^2 - 1}{(1+\eta)^2}(y^n)^2$$

and

$$L(x^n, y^{n+1}) - L(x^{n+1}, y^{n+1}) = \frac{1}{2}\frac{(1+\eta)^2 - 1}{(1+\eta)^2}(x^n)^2$$

are automatically non-negative for positive η, thus imposing no constraints. Once again this is in line with the fact that the exact solution can be reached in one step by adopting $\eta = \infty$.

3. $L = \frac{1}{2}\left(x^2 + y^2\right)$

   Here the update rule (9) yields (23), which, for $\eta > 2$, converges to $(x, y) = (0, 0)$, which is a minimum, not a min-maximizer of L. The corresponding conditions in (24) are

$$L(x^{n+1}, y^{n+1}) - L(x^{n+1}, y^n) = \frac{(y^n)^2}{2}\left(\frac{1}{(1-\eta)^2} - 1\right) \geq 0,$$

$$L(x^{n+1}, y^{n+1}) - L(x^n, y^{n+1}) = \frac{(x^n)^2}{2}\left(\frac{1}{(1+\eta)^2} - 1\right) \leq 0.$$

   While the second of these imposes no constraint, the first restricts η to be smaller than 2, thus guaranteeing divergence. This is the required output for this problem with no minimax. In a more general setting, this divergence would correspond to leaving regions surrounding critical point of the wrong type, hence opening the search for true minimax solutions elsewhere.

It can be verified that this behavior is general: in a neighborhood of a strict mini-maximizer point z^*, the learning rate η can be adopted arbitrarily large and still satisfy the constraints in (24) (see proof in Section B in the appendix). It can also be shown (see proof in Section B in the appendix) that, when the gradient is different from zero, there is always a small enough $\eta > 0$ such that the saddle point conditions (24) are satisfied. This excludes spurious convergence of the algorithm to non-critical points because no positive value of η satisfying (24) can be found.

The example $L = \frac{x^2+y^2}{2}$ above illustrates the fact that, close to a critical point of the wrong type, the constraints in (24) exclude the possibility of local convergence, except for a set of starting points with zero measure –such as the critical points themselves. This example is prototypical, in the sense that near a critical point, a smooth L can typically be approximated by a quadratic function. This behavior is further illustrated with a numerical example in two dimension in Section 7.1 where a saddle point is surrounded by four critical points that are local maxima or minima.

Thus the implicit twisted descent algorithm, complemented by the constraints in (24), converges locally to minimax points and cannot converge to critical points of the wrong type except for a set of initial values of zero measure. Of course these local results are not enough to guarantee global convergence: similarly to gradient descent, Newton or any other local procedure for regular minimization, the procedure can fail to detect a mini-maximizing point if initialized far enough from it. We will see below a simple instance of this, with the true mini-maximizer hidden by a set of regular maxima and minima that delimit its basin of attraction.

The constraints in (24) provide an upper bound for the learning rate η. Therefore a natural proposal would evolve η from step to step, making it increase –so as to approach the Newton regime– unless the constraints are not satisfied, in which case the step should be rejected and η decreased. Rather than updating the learning rate η directly, we propose to update a surrogate μ, and then build η dividing μ by $\|G^n\|^2$. The algorithm proposed is the following:

1. Set an initial guess z^0 and an initial value μ^0.
2. At each step, update μ through $\mu^{n+1} = \min\left(\alpha\mu^n, \mu_{max}\right)$, with $\alpha > 1$, $\mu_{max} \gg 1$. Update z^n to z^{n+1} through (9) with $\eta = \mu^{n+1}/\|G^n\|^2$. If the conditions in (24) are not satisfied, reduce μ^{n+1} (for instance halving it) until either they are satisfied or μ^{n+1} is smaller than a prescribed threshold.
3. Stop when either $\|G^{n+1}\|$ is smaller than a prescribed threshold or the number of steps reaches a prescribed maximum.

The reason for adopting $\eta = \mu/\|G^n\|^2$ is twofold. On the one hand, we would like η to grow fast near the optimal point, so as to yield the fast convergence associated to Newton's method. On the other, it can be shown that this normalization has the property of making η grow fast enough to yield convergence even for very flat minimax points (such as $(0,0)$ in $L = x^3y^3$), where the superlinear rate of decrease of the gradient could otherwise prevent the numerical solution from converging to the optimal point.

5 Quasi implicit twisted gradient descent

The leading computational costs of the proposed procedure are the computation of the Hessian (which may not even be available in closed form) and the inversion of its mollified version, i.e. the calculation of the matrix

$$B = \left(J + \eta H\right)^{-1}, \tag{25}$$

which becomes costly in high dimensions.

There are at least two ways to bypass the need to calculate the Hessian and B, one analogous to predictor-corrector schemes for differential equations and the other to the quasi-Newton methodology for minimization.

5.1 Predictor-corrector approach

The predictor-corrector approach starts from the fully implicit scheme in (6) and approximates it through the two-step explicit procedure

$$\begin{aligned} z^* &= z^n - \eta\ J\ G\left(z^n\right) \\ z^{n+1} &= z^n - \eta\ J\ G\left(z^*\right). \end{aligned} \tag{26}$$

For small values of $\eta\|G\|$, we can expand z^{n+1}, obtaining

$$\begin{aligned} z^{n+1} &= z^n - \eta J\ G^n + \eta^2 H^n G^n + O\left(\eta^3 \left\|G^n\right\|^2\right) \\ &= z^n - \eta \left(J + \eta H^n\right)^{-1} G^n + O\left(\eta^3 \left\|H^n\right\| \left\|G^n\right\|\right), \end{aligned}$$

proving consistency with the scheme in (9). Notice though that the consistency of (26) requires the learning rate η to be small, while (9) requires small values of $\eta\|G\|$, a much weaker constraint near critical points. Thus one could use (26) while exploring the landscape and (9) in the final stages, where η can be allowed to grow unboundedly.

5.2 Updating B

In the spirit of quasi-Newton methods [37], one can replace the B in (25) with an estimation that is updated at each time-step using our knowledge of the gradient at two consecutive times, G^n and G^{n+1}, since (15) reads:

$$JG^{n+1} = B^n G^n \tag{27}$$

plus higher order corrections. Of course, at the time of updating B^n, one does not yet know G^{n+1}. Instead, one can update B correcting B^{n-1} into a B^* that would have satisfied this constraint at the prior step:

$$JG^n = B^* G^{n-1}, \quad B^* \to B^n. \tag{28}$$

A significant difference with regular quasi-Newton methods though is that B is not positive definite, unlike the Hessian near minima. Thus, even though one could propose the equivalent to the BFGS recipe:

$$B^* = W_n^t B^{n-1} W_n + \frac{JG^n (JG^n)^t}{(G^{n-1})^t JG^n}, \tag{29}$$

where

$$W_n = I - \frac{G^{n-1}(JG^n)^t}{(G^{n-1})^t JG^n},$$

this could yield an uncontrollable large correction, since the denominator can vanish even for an arbitrarily small learning rate η, for which $B = J$ and $G^n = G^{n-1}$.

An alternative is to perform the rank-one update

$$B^n = B^* = B^{n-1} + \frac{\left(JG^n - B^{n-1}G^{n-1}\right)\left(JG^n - B^{n-1}G^{n-1}\right)^t}{(G^{n-1})^t \left(JG^n - B^{n-1}G^{n-1}\right)}.$$

Similarly to (29), this corrects B so that it satisfies (28). In order to avoid singularities when the denominator vanishes, we may write this as

$$B^* = B^{n-1} + \alpha \frac{\left(JG^n - B^{n-1}G^{n-1}\right)\left(JG^n - B^{n-1}G^{n-1}\right)^t}{\left\|(JG^n - B^{n-1}G^{n-1}\right\|^2}, \tag{30}$$

with

$$\alpha = \frac{\left\|(JG^n - B^{n-1}G^{n-1}\right\|^2}{(G^{n-1})^t\left(JG^n - B^{n-1}G^{n-1}\right)}$$

replaced with

$$\alpha^* = \operatorname{sign}(\alpha)\min\left(|\alpha|, \gamma\right), \quad \gamma < 1.$$

i.e. bounding the Frobenius norm of the rank-one update. This is not a costly bound to impose: once near convergence, z changes little in each step –since $\|G\|$ is small– so B requires only a bounded update per step. Applying a similar solution to eliminate possible singularities to (29) would have been problematic, as we would have had to fix not only the second term of the sum on the right hand side of (29) but also the matrix W_n.

The reason for picking $|\alpha^*| \le \gamma < 1$ is the following. Assume without loss of generality that the optimal $z = z^*$ equals zero and that the true Hessian at $z^* = 0$ is H. Then, for z^n close to z^*, we have that $G \approx Hz$ and

$$\begin{aligned} z^{n+1} &= z^n - \eta\left[B^{n+1} - \left(B^{n+1} - B^n\right)\right]G^n \\ &= z^n - \eta JG^{n+1} + \eta\left(B^{n+1} - B^n\right)G^n \quad \text{(from (28))} \\ &\approx z^n - \eta JHz^{n+1} + \eta\left(B^{n+1} - B^n\right)Hz^n, \end{aligned}$$

so

$$z^{n+1} \approx (I + \eta JH)^{-1}\left[I + \eta\left(B^{n+1} - B^n\right)H\right]z^n.$$

Then $w = Hz \approx G$ satisfies

$$w^{n+1} \approx H(I + \eta JH)^{-1}\left[H^{-1} + \eta\left(B^{n+1} - B^n\right)\right]w^n.$$

But one can readily show that

$$H(I + \eta JH)^{-1} = (I + \eta HJ)^{-1}H,$$

so

$$w^{n+1} \approx (I + \eta HJ)^{-1}\left[I + \eta H\left(B^{n+1} - B^n\right)\right]w^n,$$

or

$$w^{n+1} \approx (I + \eta\tilde{H})^{-1}\left[I + \eta\tilde{H}J\left(B^{n+1} - B^n\right)\right]w^n,$$

with $\tilde{H} = HJ$. We can bound the norm of the sum within brackets using the singular value of the rank-one update for B:

$$\|w^{n+1}\| \le \left\|(I + \eta\tilde{H})^{-1}\left[I + \eta|\alpha^*|\tilde{H}\right]w^n\right\|.$$

Under the hypothesis in Theorem 1 or the less restrictive Theorem 3 in Appendix A, $\tilde{H}$ is non-negative definite and invertible. It follows that

$$|\alpha^*| \leq \gamma < 1 \Rightarrow \|w^{n+1}\| < \|w^n\|,$$

i.e. under the chosen bound for α^*, the gradient of L decreases at each step, and the quasi-Newton algorithm converges.

In order to turn this argument into a proof, we would need to show that it is indeed possible to satisfy (28) when the rank-one update in (30) is bounded by γ. Intuitively this follows from the fact that, near convergence, the true Hessian changes little at each step, so the required updates are necessarily small.

One extra consideration is that, if η needs to be decreased significantly within one step so as to satisfy the constraints in (24), the estimated B should converge to J. This can be achieved through the correction

$$B \rightarrow \alpha B + (1-\alpha) J,$$

with a factor α that converges to zero as η does, such as

$$\alpha = \frac{\eta}{\eta_0},$$

where η_0 is the first learning rate attempted at the current step.

6 Inequality constraints

Often some or all z_i are required to be in some subset, typically to be non-negative. We can limit consideration to this latter case with little loss of generality, since any constraint of the form $g(z) \geq 0$ can be reduced to the positivity of the corresponding Lagrange multiplier. So we have the problem

$$\min_x \max_y L(x, y), \quad x(P_x) \geq 0, \quad y(P_y) \geq 0,$$

where P_x and P_y index the subset of variables required to be non-negative. There are a number of ways to extend the procedure of this article to the case with inequalities; we discuss below two alternative methodologies:

6.1 Change of variables

The simplest way to enforce positivity without altering the algorithm is to make a change of variables that ensures positivity, for instance setting

$$L^*(x, y) = L(X(x), Y(y)),$$

where

$$X(x) = \begin{cases} x & \text{for unrestricted variables} \\ x^2 & \text{for variables required to be non-negative} \end{cases} \tag{31}$$

and similarly for $Y(y)$. This yields the unconstrained minimax problem

$$\min_x \max_y L^*(x, y)$$

to which the procedure can be applied, and whose solution, once transformed into (X, Y), solves the original problem

$$\min_X \max_Y L(X, Y), \quad X(P_x) \geq 0, Y(P_y) \geq 0.$$

A word of caution is in order though: the fact that, for $i \in P_x$, we have that $x_i = 0 \Rightarrow L^*_{x_i} = 0$, and similarly for y_j, creates potential suboptimal points where the procedure might stop. For instance, in constrained optimization problems, the Lagrange multipliers corresponding to inactive constraints are zero at the solution, but one often encounters along the way to the true solution, domains where some constraints that will be active in the final solution are temporarily inactive. Hence these Lagrange multipliers z_i may reach machine zero values, at which point the corresponding derivatives of L^* vanish. Because of this, these z_i may fail to leave zero when the corresponding constraints become active again.

This issue can be addressed through a simple procedural change: after every step, compute the gradient of the original Lagrangian, i.e. L_Z, for the variables $\{Z_i\}$ that are close to zero, i.e. $\|z_i\| \leq \epsilon$. Since z_i should detach from zero when the original gradient L_{Z_i} pushes $Z_i = {z_i}^2$ to be positive, we compute

$$R_i = \max(-J_i^i L_{Z_i}, 0)$$

and update z via

$$z_i \to \tilde{z}_i = \sqrt{{z_i}^2 + \eta_0 R_i}, \tag{32}$$

where η_0 is a suitably small additional learning rate, restricted so as to satisfy the requirements in (24) between the states z_i and $\tilde{z}_i$. To do this, we start with an arbitrary value for η_0 and reduce it, for instance by halving, until (24) is satisfied.

6.2 Evolving barriers

A more conventional approach to handling positivity constraints is to add a logarithmic barrier:

$$L(x, y) \to L^t(x, y) = L(x, y) + \frac{1}{t}\left[\sum_j \log(y_j) - \sum_i \log(x_i)\right]. \tag{33}$$

Here we can either solve the problem for an increasing sequence of values of t, adopting as initial values of (x, y) for each subproblem their terminal values from the prior one, or take this to the limit, evolving t smoothly at each step of the algorithm.

7 Examples

We illustrate the procedure through three types of examples: some simple two-dimensional problems designed to illustrate the effects of non-convexity on quasi-implicit descent and the need for the constraints imposed on the learning rate, a linear programming problem that illustrates the handling of inequality constraints when very many are simultaneously active, and an optimal transport problem to show a nonlinear adversarial example of current interest.

7.1 Two-dimensional examples

This sub-section displays numerical examples of the implicit gradient descent and the quasi Newton method on non-monotone saddle point problems (i.e. one in which the objective function is not convex-concave in the variables in which we are minimizing and maximizing respectively). The first Lagrangian we consider is

$$L(x, y) = (x - 0.5)(y - 0.5) + \frac{1}{3} e^{(-(x-0.5)^2-(y-0.75)^2)}. \tag{34}$$

This function has a saddle point near $(0.5, 0.5)$ and a local maximum near $(0.5, 0.75)$. It has been observed in [29] that, in this case, first-order descent methods result in periodic orbits. Figure 1 shows that this is indeed the case if the look ahead time η in (9) is very small, effectively reducing (9) to an explicit algorithm. It is also interesting to notice that, in line with the discussion in section 3, for large values of η we reach a very fast convergence since, close to the saddle point of $f(x, y)$, the implicit (9) is essentially exact.

Figure 2 shows the performance of the Quasi Newton algorithm with variable learning rates η of section (5). We see that the algorithm effectively "learns" the Hessian, leading to convergence. The jumps of the value of η correspond to violation of the constraint in (24).

Next consider an example of escape from local maxima or minima given by the function

$$L(x, y) = xy\ e^{-\frac{(x^2+y^2)}{2}}.$$

This function has a saddle point at $(0, 0)$ and local maxima and minima at the four corners of a square $\mathcal{S}$, with edge length of 2, centered at the origin. Figure 3 shows the L with the trajectory followed by our algorithm (9) with initial point at $(1 - 10^{-5}, -0.2)$ and initial value of $\eta = 0.5$. The trajectory initially climbs the local maxima in $(1, 1)$ along a direction where the gradient is essentially zero in the x direction, as a result the saddle point conditions (24) are satisfied and η increases. When the trajectory approaches the local maxima

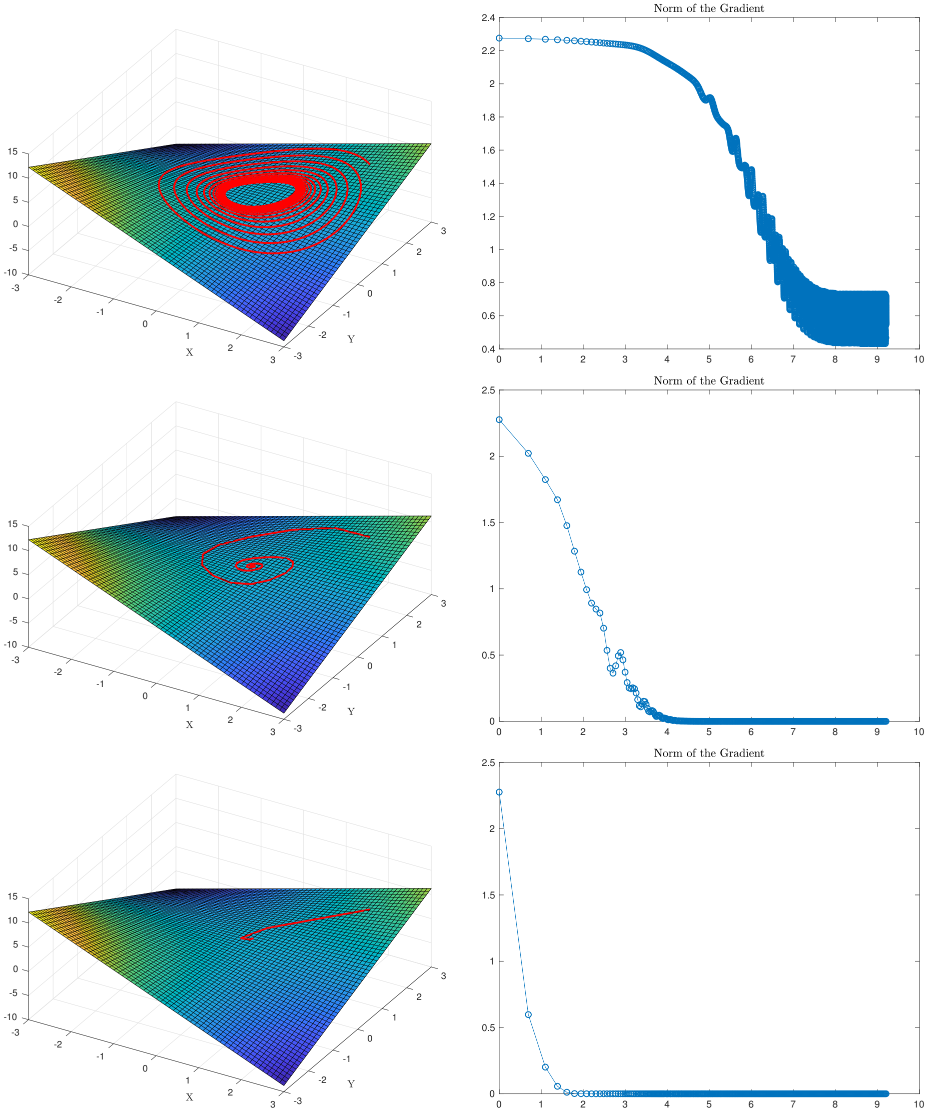

Fig. 1 Three trajectories using the algorithm in (9) with 3 different values of fixed η to compute the saddle point of (34). The left column shows the trajectory and the right column the value of the norm of the gradient appearing in (9) as a function of the logarithm of the iteration step. Each row of the plot is obtained with values of η equal to 0.05, 0.5 and 5 respectively. For a too small value of the "looking forward" time η the algorithm behaves essentially as the analogous gradient ascent-descent resulting in a periodic orbit. As the value of η increases the gradient decreases as described by (16).

y it becomes harder and harder to satisfy the conditions (24) until, in order to satisfy them η has to decreases making (9) essentially an explicit scheme that now follows the gradient (see Figure 3) pointing to the local minima in

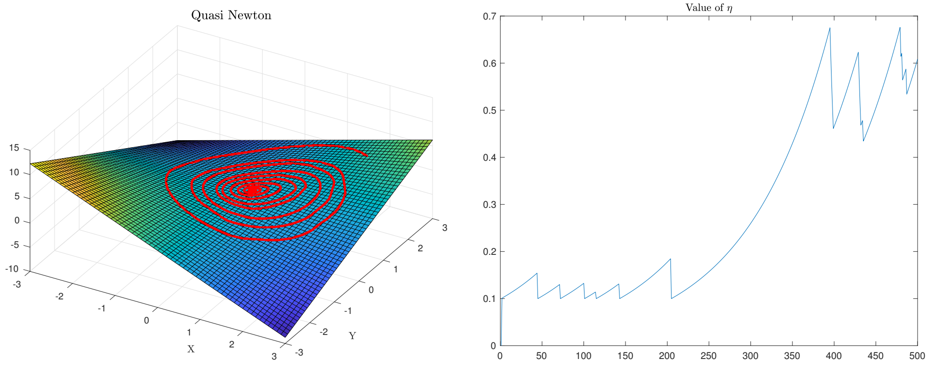

Fig. 2 Trajectory obtained when using the Quasi Newton algorithm with variable η as described in section 5. It can be seen that the learning rate η get smaller in certain points of the trajectory due to the enforcing of the conditions in (24).

$(-1, 1)$. This behavior is repeated until the trajectory is sufficiently close to the origin so that η can keep increasing without violating (24) and the algorithm essentially becomes a Newton method.

This example shows the local nature of the convergence of the scheme in (9). With initial position $1, -0.2$ the gradient in x is exactly zero and the trajectory converges to the local maxima in $(1, 1)$. If instead the initial condition where $(1 + \delta, -0.2)$ then the trajectory would escape the local maxima at $(1, 1)$ by leaving $\mathcal{S}$ and following a direction in which y is constant and x increased.

7.2 Linear programming

We consider the standard linear programming problem

$$\min_{X \geq 0} c^t X, \quad AX \geq b, \tag{35}$$

which, introducing Lagrange multipliers Y for the constraints, adopts the Lagrangian form

$$\min_{X \geq 0} \max_{Y \geq 0} \; L^*(X, Y) = c^t X - Y^t (AX - b) \,. \tag{36}$$

To eliminate the positivity constraints, we introduce unconstrained variables x and y through $X = x^2$, $Y = y^2$, both understood component-wise, which yields the unconstrained minimax problem

$$\min_x \max_y \; L(x, y) = c^t X(x) - Y(y)^t (AX(x) - b) \,. \tag{37}$$

We have

$$L_x = 2 \left(c - A^t Y \right) .* x, \quad L_y = 2 \left(b - AX \right) .* y, \tag{38}$$

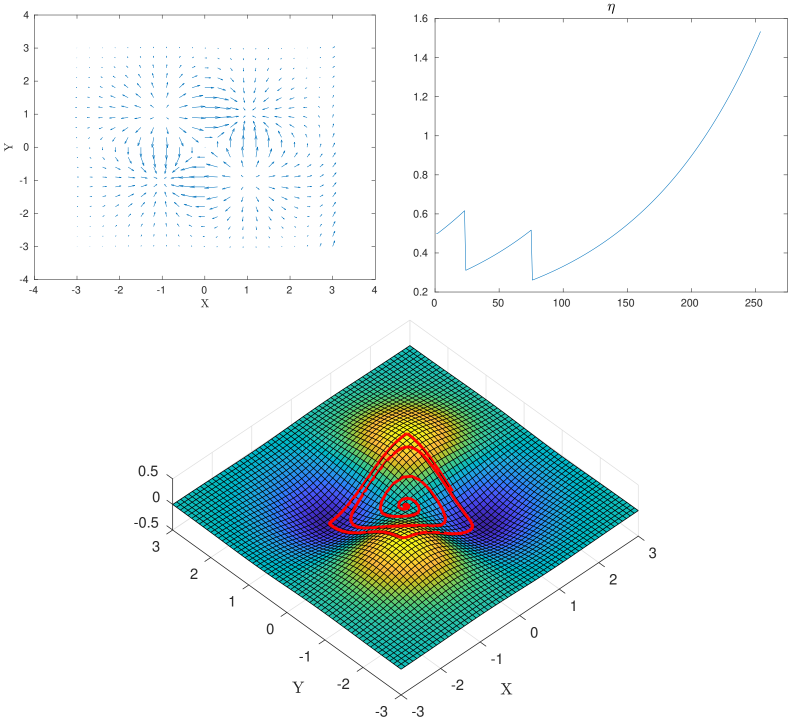

Fig. 3 Escape from critical points of the wrong type. Upper left: Gradient field of L. Upper right: η as a function of the iteration. Lower panel: Trajectory starting at $(1-10^{-5}, -0.2)$ and converging to the saddle point.

and

$$\begin{aligned} L_{xx} &= 2 \text{ diag}\left(c - A^t Y\right), & L_{xy} &= -4 \text{ diag}(x) A^t \text{diag}(y) \\ L_{yx} &= -4 \text{ diag}(y) A \text{ diag}(x) & L_{yy} &= 2 \text{ diag}\left(b - AX\right). \end{aligned}$$

where the symbol '.$*$' denotes component-wise multiplication, and 'diag(x)' denotes a diagonal matrix with the vector x on its diagonal.

For the example displayed in Figure 4, we chose $n_x = 117$, $n_y = 114$. All entries of the matrix A and the vectors b and c were drawn independently from the uniform distribution in $[0, 1]$, thus guaranteeing feasibility.

The only free parameters of the procedure are the maximum learning rate, which we fixed at 10^7, the rate $\alpha = 5.1$ at which μ is updated, and the initialization of x and y, for which we picked quite arbitrarily

$$x_0(1:n_x) = \sqrt{\frac{0.8}{n_x}}, \quad y_0(1:n_y) = \sqrt{\frac{0.4}{n_y}}.$$

For every realization of the problem, the procedure converges invariably to the right answer in 200-300 steps. A characteristic of this problem is that

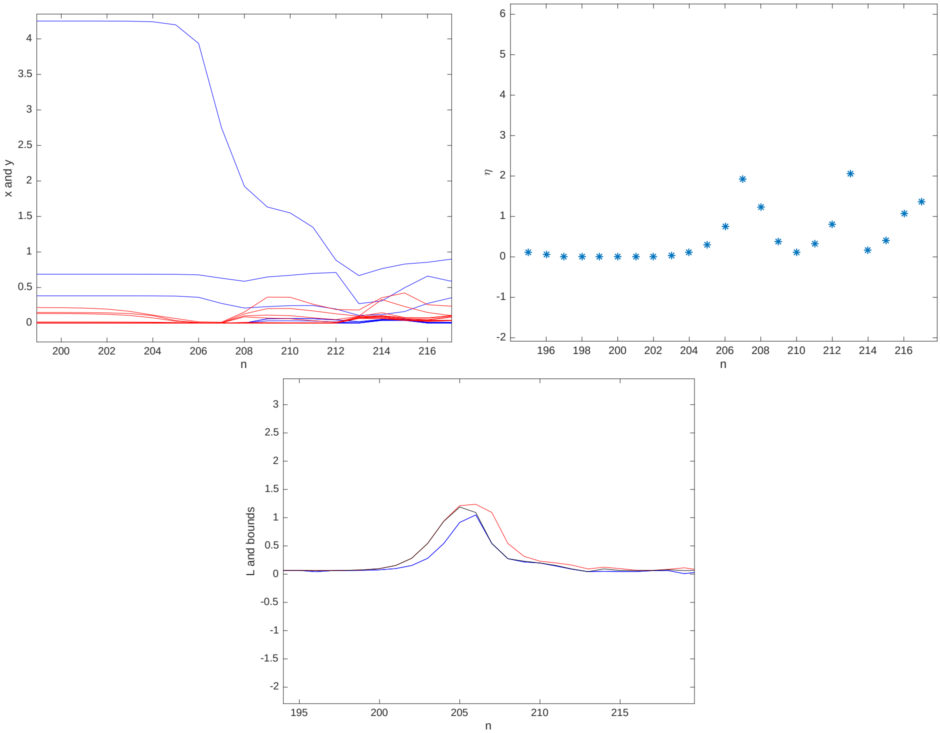

Fig. 4 Linear Programming, zoom of the evolution near times ($n = 206, 212$) where the active set changes considerably. Upper left panel: evolution of the 10 largest x_i (in blue) and y_j (in red). Upper right panel: learning rate η. Lower panel: the three values of L appearing in the checks in (24), with the actual future L in black and its upper and lower bounds in red and blue respectively.

most positivity constraints are active, not only in the final solution but also at intermediate steps. Figure 4 displays the working of the procedure at times where the active set changes significantly. We can see a local increase of the learning rate, corresponding to the opening of a significant gap between the lower and upper bounds for L in (24).

7.3 Optimal Transport

An adaptive, adversarial methodology was developed in [22] for the optimal transport problem [38, 39], between two distributions μ and ν, known only through a finite set of independent samples. The problem consists in finding a global map T, pushing samples generated by the source μ, so that their final distribution matches ν, the one underlying the samples of the target. In addition, this map should minimize a transportation cost. For quadratic cost functions, the map T must be given by the gradient $\nabla\phi$ of a convex potential ϕ. We generate T by composing many elementary non-linear functions u_k. Each of these u_k minimizes a local optimal transport problem between two nearby samples $(x_i^{(k)})_{i=1,\ldots,n}$ and $(y_j^{(k)})_{j=1,\ldots,m}$. A global iterative procedure using displacement interpolation guarantees convergence to the unique optimizer.

In order to find these local non-linear maps u, we minimize the Kullback-Leibler divergence between the distributions underlying $u(x_i)$ and y_j. A variational characterization of the Kullback-Leibler divergence gives rise to the following formulation of the local problem:

$$\min_{u=\nabla\phi} \max_{g} \left\{ \frac{1}{n} \sum_i g(u(x_i)) - \frac{1}{m} \sum_j e^{g(y_j)} \right\} \tag{39}$$

The above mini-maximization can be interpreted as a two player game between the map u and the lens g: as u does its best to push the x_i's toward the y_j's, g will focus on the areas where the mass transport has not yet been well achieved. This forces u to correct those areas, and g to find new locations requiring more work.

The maps u and g are parameterized using finite dimensional vectors α and β, and the problem is reduced to:

$$\min_{\alpha} \max_{\beta} \left\{ \frac{1}{n} \sum_i g_\beta(u_\alpha(x_i)) - \frac{1}{m} \sum_j e^{g_\beta(y_j)} \right\} \equiv \min_{\alpha} \max_{\beta} L(\alpha, \beta) \tag{40}$$

We solve each of those local optimal problems using the methodology described in this manuscript.

Figure 5 presents the original configuration of samples and the result of the global procedure, applied to data $\{x_i\}$ drawn from a Gaussian and $\{y_j\}$ from the uniform distribution on the perimeter of a circle. Figure 6 displays the objective function at each step, for the last local optimal transport problem of the first global iteration. In addition to $L(\alpha^n, \beta^n)$, displayed in orange, the

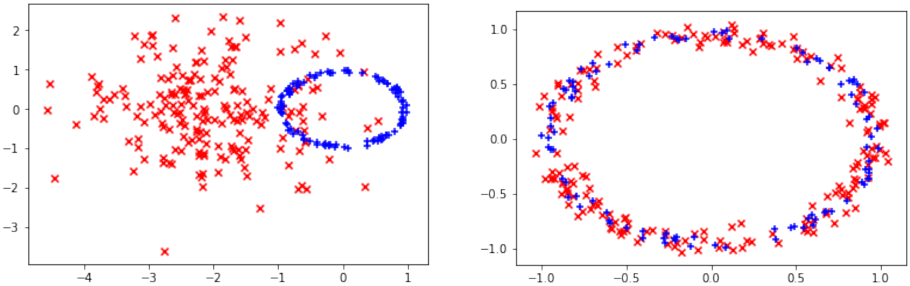

Fig. 5 Initial and final configuration of the global optimal transport algorithm. The blue crosses represent the samples (y_j), the red crosses on the left figure represent the samples (x_i), and the red crosses in the right figure represent the samples generated by $T(x_i)$ where T is a solution of the optimal transport algorithm

upper bound $L(\alpha^n, \beta^{n+1})$ and the lower bound $L(\alpha^{n+1}, \beta^n)$ are displayed in green and blue respectively.

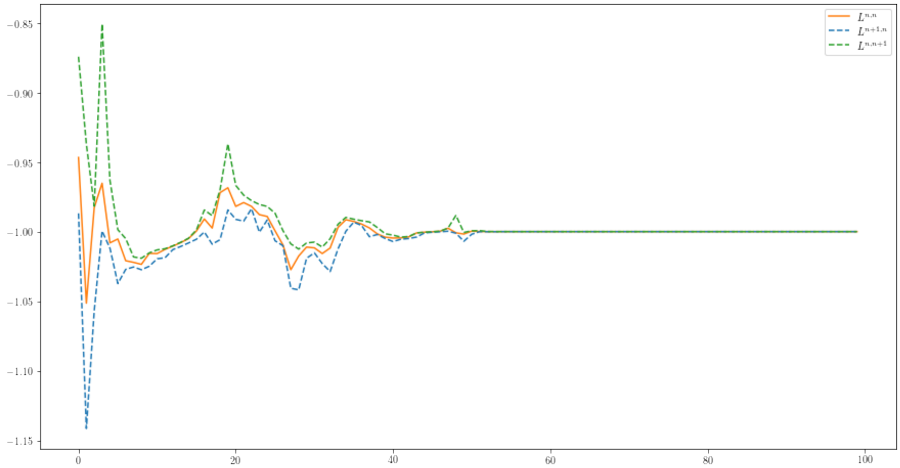

Fig. 6 Values of the Lagrangian at $L(\alpha^n, \beta^n) \equiv L^{n,n}, L(\alpha^{n+1}, \beta^n) \equiv Ln+1, n$ and $L(\alpha^n, \beta^{n+1}) \equiv L^{n,n+1}$ for the last local optimal transport problem of the first global iteration.

8 Conclusions

This article presents an implicit twisted gradient descent strategy for the numerical computation of saddle points. Explicit methods are by nature non-anticipatory, which makes them often fail to converge, ending out in periodic or outward spiraling orbits around a saddle point. Instead, the algorithm proposed here is implicit, or anticipatory from a game theory perspective, as each player includes their adversary's best strategy in their own planning. This is proved to yield local convergence, which acquires a super-quadratic rate as the learning rate grows and the methodology converges to Newton's. The strategy proposed for updating the learning rate is consistent with the anticipatory nature of the algorithm: the rate should grow rapidly near saddle points, but is bounded by the requirement that, given the adversary's choice, each player should be improving their game. This guarantees convergence near saddle points and local divergence near stationary points of the Lagrangian that do not solve the minimax problem, points toward which regular Newton would otherwise converge.

The use of an implicit algorithm requires the inversion of a matrix, which can be quite large for high-dimensional problems. To alleviate this computational cost, the analogue of a quasi-Newton formulation of the algorithm is developed, which updates directly the inverse B of the mollified Hessian at the core of the algorithm. This not only serves the purpose of avoiding matrix inversion, but also eliminates the need to compute or estimate second derivatives of the Lagrangian.

Numerical tests are performed on three representative problems: a small-dimensional minimax problem that does not satisfy global convex-concavity, linear programming with a high number of inequality constraints, and a recently proposed adversarial methodology for optimal transport. In their diversity, they illustrate the versatility of the proposed methodology, which can be applied without modifications to virtually any minimax problem. It has been the author's experience that having such a general tool at one's disposal encourages the formulation of problems of interest in adversarial terms, a natural characterization that one would otherwise often avoid for lack of a straightforward methodology for their numerical solution.

9 Acknowledgements

The work of E. G. Tabak was partially supported by NSF grant DMS-1715753 and ONR grant N00014-15-1-2355. The work of M. Essid was partially supported by NSF grant DMS-1311833.

A Sufficient conditions for strict local mini-maximizers

Theorem 3 *The following conditions guarantee that $z^* = (x^*, y^*)$ is a strict local mini-maximizer of a smooth function $L(x, y)$:*

1. *The conditions in (10) are satisfied. This implies in particular that L_x and L_y vanish and that L_{xx} is positive semi-definite and L_{yy} is negative semi-definite.*
2. *The null spaces of L_{xx} and L_{yx} intersect only at the zero vector of X, and the null spaces of L_{yy} and L_{xy} only at the zero vector of Y.*

Proof In order to extend the proof of theorem 1 so that it applies to the newly relaxed hypotheses, notice that two subspaces A and B of a finite-dimensional vector space V intersect only at the zero vector of V if and only if there exists a number $\beta < 1$ such that

$$x \in A \;\text{ and }\; y \in B \;\Rightarrow |\langle x, y\rangle| \le \beta \|x\| \|y\|.$$

Here β quantifies how *oblique* the two subspaces are to each other, with $\beta = 0$ when $A \perp B$ and β approaching 1 when there exist vectors $x \in A$ almost parallel to vectors $y \in B$. Thus the hypotheses in *2.* can be restated as the existence of a number $\beta < 1$ such that

$$x_1 \in N_{xx} \;\text{ and }\; x_3 \in N_{yx} \;\Rightarrow |\langle x_1, x_3\rangle| \le \beta \|x_1\| \|x_3\|$$

and

$$y_1 \in N_{yy} \;\text{ and }\; y_3 \in N_{xy} \;\Rightarrow |\langle y_1, y_3\rangle| \le \beta \|y_1\| \|y_3\|.$$

The fact that the conditions in (10) require L_x and L_y to vanish, and L_{xx} and L_{yy} to be at least positive and negative semi-definite respectively follows from the same argument as in theorem 1. We will also use the same definitions as in theorem 1 for the constants s_{xx}, s_{yx}, s_{yy}, S_{xx}, S_{yx}, and S_{yy}, and write

$$\Delta x = x - x^* = x_1 + x_2 = x_3 + x_4,$$

where

$$x_1 \in N_{xx}, \quad x_2 \in N_{xx}^\perp, \quad x_3 \in N_{yx}, \quad x_4 \in N_{yx}^\perp.$$

The fact that N_{xx} and N_{yx} are oblique implies that

$$\max\left(\|x_2\|^2, \|x_4\|^2\right) \geq \gamma \|\Delta x\|^2, \tag{41}$$

where

$$\gamma = \left(\frac{1-\beta}{2}\right)^2,$$

as follows from the following argument:

$$\|x_2\|^2 \leq \gamma\|\Delta x\|^2 \Rightarrow \|x_1\|^2 \geq (1-\gamma)\|\Delta x\|^2,$$

since $\|x_1\|^2 + \|x_2\|^2 = \|\Delta x\|^2$. But

$$\|x_1\|^2 \geq (1-\gamma)\|\Delta x\|^2 \Rightarrow \|x_4\|^2 \geq \gamma\|\Delta x\|^2,$$

since

$$\begin{aligned} \|x_1\|^2 &= \langle x_1, x_1 + x_2\rangle = \langle x_1, x_3 + x_4\rangle \\ &\leq \beta\|x_1\|\|x_3\| + \|x_1\|\|x_4\| \\ &\leq \beta\|\Delta x\|^2 + \|x_1\|\|x_4\|, \end{aligned}$$

from which it follows that

$$\|x_1\|^2 \geq (1-\gamma)\|\Delta x\|^2 \Rightarrow \frac{1-\gamma-\beta}{1-\gamma} \leq \frac{\|x_4\|}{\|\Delta x\|},$$

so

$$\|x_4\|^2 \geq \left(\frac{1-\gamma-\beta}{1-\gamma}\right)^2 \|\Delta x\|^2 \geq \gamma\|\Delta x\|^2,$$

completing the proof of (41).

This inequality allows us to consider two scenarios:

1. If $\|x_2\|^2 \geq \gamma\|\Delta x\|^2$, we can set $\Delta y = 0$, and obtain

$$\begin{aligned} L(x,y) &= L(x^*,y^*) + \frac{1}{2}\Delta x^t L_{xx}\Delta x + \; O\left(\|\Delta x\|^3\right) \\ &= L(x^*,y^*) + \frac{1}{2}x_2^t L_{xx} x_2 + \; O\left(\|\Delta x\|^3\right) \\ &\geq L(x^*,y^*) + \frac{\gamma^2 s_{xx}}{2}\|\Delta x\|^2 + \; O\left(\|\Delta x\|^3\right) \\ &> L(x^*,y^*) \end{aligned}$$

   for small enough $\|\Delta x\|$, as required by condition (11).
2. If $\|x_2\|^2 \leq \gamma\|\Delta x\|^2$, it follows from (41) that $\|x_4\|^2 \geq \gamma\|\Delta x\|^2$, and the proof follows exactly the same path as in theorem 1.

The same argument, mutatis mutandis, proves condition (12).

□

In fact, for quadratic Lagrangians, these more general sufficient conditions are also necessary: if the null spaces of L_{xx} and L_{yx} share a nonzero vector a, we can adopt $\Delta x = x - x^* = a$, which yields

$$\forall y \; L(x, y) - L\left(x^*, y^*\right) = \frac{1}{2}\Delta y^t L_{yy} \Delta y \le 0,$$

contradicting the condition in (11). The same argument applies if the null spaces of L_{yy} and L_{xy} share a nonzero vector b.

B Existence of a compatible learning rate η

This section answers two related questions: whether one can always find a small-enough learning rate η so that the conditions in (24) are satisfied, and whether, in a neighborhood of a strict mini-maximizer point z^*, a learning rate η of order one or larger could be adopted, to guarantee that theorem 2 applies. The answer to both questions is affirmative, as the following arguments show.

Theorem 4 *Given a point (x, y) such that L_x and L_y are different from zero, it is always possible to find a small enough $\eta > 0$ such that (24) are satisfied.*

Proof In the limit of small η we rewrite (9) as

$$z^{n+1} = z^n - \eta J (I - \eta H J)^{-1} G = z^n + \eta J (I + \eta H J) G + O(\eta^3)$$

Without loss of generality we can assume $n = 0$ and $z^0 = 0$. Then the scheme above can be rewritten as

$$\begin{pmatrix} x^1 \\ y^1 \end{pmatrix} = -\eta \begin{pmatrix} L_x \\ L_y \end{pmatrix} + \eta^2 \begin{pmatrix} L_{xx} L_x - L_{xy} L_y \\ -L_{yx} L_x + L_{yy} L_y \end{pmatrix} + O(\eta^3) \tag{42}$$

where, when no otherwise specified, all the first and second derivatives of L are computed at $(x = 0, y = 0)$. With $L^{01} = L(x^0, y^1)$, $L^{11} = L(x^1, y^1)$ and $L^{10} = L(x^1, y^0)$, (24) becomes $L^{10} \le L^{11} \le L^{01}$. In the limit of small $\Delta x = x^1 - x^0$ and $\Delta y = y^1 - y^0$, we have

$$L^{11} = L^{00} + L_x^t \Delta x + L_y^t \Delta y + \frac{\Delta x^t L_{xx} \Delta x}{2} + \frac{\Delta y^t L_{yy} \Delta y}{2} + O(\Delta^3) \tag{43}$$

$$L^{10} = L^{00} + L_x^t \Delta x + \frac{\Delta x^t L_{xx} \Delta x}{2} + O(\Delta^3) \tag{44}$$

$$L^{01} = L^{00} + L_y^t \Delta y + \frac{\Delta y^t L_{yy} \Delta y}{2} + O(\Delta^3). \tag{45}$$

We will consider only $L^{10} \le L^{11}$, as the other side of (24) can be treated analogously. With the expansions written above, $L^{10} \le L^{11}$ becomes $0 \le L^{00} + L_x^t \Delta x + + \frac{\Delta x^t L_{xx} \Delta x}{2} + \Delta x^t L_{xy} \Delta y$, which after substituting the expressions for Δx and Δy from (42) becomes

$$\eta \|L_y\|^2 + \frac{3}{2}\eta^2 \Delta x^t L_{xx} \Delta x + O(\eta^3) \ge 0,$$

a condition that can always be satisfied for sufficiently small η when the norm of L_y is different from zero. □

Theorem 5 *Given a local saddle point (x, y) there exists a neighborhood (U_x, U_y) such that $\forall (x, y) \in (U_x, U_y)$ the condition (24) is satisfied for an arbitrarily large value of η*

Proof Consider one side of (24) (the other side can be deduced analogously):

$$0 \leq L(x^{n+1}, y^{n+1}) - L(x^{n+1}, y^n) = \Delta y^t \left[L_y^n + L_{xy}^n \Delta x \right] + \frac{\Delta y^t L_{yy}^n \Delta y}{2} + O(\Delta^3) \quad (46)$$

where $\Delta y = y^{n+1} - y^n$. The term in square brackets on the right hand side of (46) can be rewritten using the component of (9) relative to the variable y:

$$\Delta y^t \left[L_y^n + L_{yx}^n \Delta x \right] = \frac{\|\Delta y\|^2}{\eta} - \Delta y^t L_{yy}^n \Delta y, \quad (47)$$

which, substituted in (46), leads to

$$\frac{\|\Delta y\|^2}{\eta} - \frac{\Delta y^t L_{yy}^n \Delta y}{2} \geq 0,$$

a condition that holds for every positive η, given that L_{yy} must be negative semi-definite at a maximum. □

Data Availability

The datasets generated and analysed during the current study are not publicly available due the fact that they can be easily recreated using the information contained in the text. If needed the corresponding author is available for clarification on reasonable request.

References

[1] Von Neumann, J.: Zur Theorie der Gesellschaftsspiele, in Mathematische Annalen, 100. Julius Springer Berlin (1928)

[2] Morgenstern, O., Von Neumann, J.: Theory of Games and Economic Behavior. Princeton university press, ??? (1953)

[3] Benzi, M., Golub, G.H., Liesen, J.: Numerical solution of saddle point problems. Acta numerica **14**, 1–137 (2005)

[4] Schöberl, J., Zulehner, W.: Symmetric indefinite preconditioners for saddle point problems with applications to pde-constrained optimization problems. SIAM Journal on Matrix Analysis and Applications **29**(3), 752–773 (2007)

[5] Gerner, A.-L., Veroy, K.: Certified reduced basis methods for parametrized saddle point problems. SIAM Journal on Scientific Computing **34**(5), 2812–2836 (2012)

[6] Mokhtari, A., Ozdaglar, A., Jadbabaie, A.: Escaping saddle points in constrained optimization. Advances in Neural Information Processing Systems **31** (2018)

[7] Angot, P., Caltagirone, J.-P., Fabrie, P.: A new fast method to compute saddle-points in constrained optimization and applications. Applied Mathematics Letters **25**(3), 245–251 (2012)

[8] Shafieezadeh Abadeh, S., Mohajerin Esfahani, P.M., Kuhn, D.: Distributionally robust logistic regression. Advances in Neural Information Processing Systems **28** (2015)

[9] Pfau, D., Vinyals, O.: Connecting generative adversarial networks and actor-critic methods. arXiv preprint arXiv:1610.01945 (2016)

[10] Sinha, A., Namkoong, H., Volpi, R., Duchi, J.: Certifying some distributional robustness with principled adversarial training. ICLR (2018)

[11] Palaniappan, B., Bach, F.: Stochastic variance reduction methods for saddle-point problems. Advances in Neural Information Processing Systems **29** (2016)

[12] Tabak, E.G., Trigila, G., Zhao, W.: Conditional density estimation and simulation through optimal transport. Machine Learning **109**(4), 665–688 (2020)

[13] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)

[14] Santambrogio, F.: Optimal transport for applied mathematicians. Birkäuser, NY **55**(58-63), 94 (2015)

[15] Dan, Y., Zhao, Y., Li, X., Li, S., Hu, M., Hu, J.: Generative adversarial networks (gan) based efficient sampling of chemical composition space for inverse design of inorganic materials. npj Computational Materials **6**(1), 1–7 (2020)

[16] Vogt, M.: Using deep neural networks to explore chemical space. Expert Opinion on Drug Discovery **17**(3), 297–304 (2022)

[17] Kimmel, J.C., Kelley, D.R.: Semisupervised adversarial neural networks for single-cell classification. Genome research **31**(10), 1781–1793 (2021)

[18] Barile, B., Marzullo, A., Stamile, C., Durand-Dubief, F., Sappey-Marinier, D.: Data augmentation using generative adversarial neural networks on

brain structural connectivity in multiple sclerosis. Computer methods and programs in biomedicine **206**, 106113 (2021)

[19] Alqahtani, H., Kavakli-Thorne, M., Kumar, G.: Applications of generative adversarial networks (gans): An updated review. Archives of Computational Methods in Engineering **28**(2), 525–552 (2021)

[20] Galichon, A.: Optimal Transport Methods in Economics. Princeton University Press, ??? (2018)

[21] Carlier, G., Oberman, A., Oudet, E.: Numerical methods for matching for teams and wasserstein barycenters. ESAIM: Mathematical Modelling and Numerical Analysis **49**(6), 1621–1642 (2015)

[22] Essid, M., Laefer, D.F., Tabak, E.G.: Adaptive optimal transport. Information and Inference: A Journal of the IMA **8**(4), 789–816 (2019)

[23] Tabak, E.G., Trigila, G., Zhao, W.: Distributional barycenter problem through data-driven flows. Pattern Recognition, 108795 (2022)

[24] Kobyzev, I., Prince, S.J., Brubaker, M.A.: Normalizing flows: An introduction and review of current methods. IEEE transactions on pattern analysis and machine intelligence **43**(11), 3964–3979 (2020)

[25] Trigila, G., Tabak, E.G.: Data-driven optimal transport. Communications on Pure and Applied Mathematics **69**(4), 613–648 (2016)

[26] Tabak, E.G., Turner, C.V.: A family of non-parametric density estimation algorithms. CPAM **LXVI** (2013)

[27] Kose, T.: Solutions of saddle value problems by differential equations. Econometrica, Journal of the Econometric Society, 59–70 (1956)

[28] Uzawa, H.: Iterative methods for concave programming. Studies in linear and nonlinear programming **6**, 154–165 (1958)

[29] Mertikopoulos, P., Zenati, H., Lecouat, B., Foo, C.-S., Chandrasekhar, V., Piliouras, G.: Mirror descent in saddle-point problems: Going the extra (gradient) mile. arXiv preprint arXiv:1807.02629 (2018)

[30] Adolphs, L., Daneshmand, H., Lucchi, A., Hofmann, T.: Local saddle point optimization: A curvature exploitation approach. In: The 22nd International Conference on Artificial Intelligence and Statistics, pp. 486–495 (2019). PMLR

[31] Mazumdar, E.V., Jordan, M.I., Sastry, S.S.: On finding local nash equilibria (and only local nash equilibria) in zero-sum games. arXiv preprint arXiv:1901.00838 (2019)

[32] Rafique, H., Liu, M., Lin, Q., Yang, T.: Weakly-convex–concave min–max optimization: provable algorithms and applications in machine learning. Optimization Methods and Software, 1–35 (2021)

[33] Essid, M., Tabak, E., Trigila, G.: An implicit gradient-descent procedure for minimax problems. arXiv preprint arXiv:1906.00233 (2019)

[34] Holding, T., Lestas, I.: On the convergence to saddle points of concave-convex functions, the gradient method and emergence of oscillations. In: 53rd IEEE Conference on Decision and Control, pp. 1143–1148 (2014). IEEE

[35] Nemirovski, A.: 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 Journal on Optimization **15**(1), 229–251 (2004)

[36] Du, S.S., Hu, W.: Linear convergence of the primal-dual gradient method for convex-concave saddle point problems without strong convexity. In: The 22nd International Conference on Artificial Intelligence and Statistics, pp. 196–205 (2019). PMLR

[37] Nocedal, J.: Updating quasi-newton matrices with limited storage. Mathematics of computation **35**(151), 773–782 (1980)

[38] Monge, G.: Mémoire sur la Théorie des Déblais Et des remblais. De l'Imprimerie Royale, ??? (1781)

[39] Kantorovich, L.V.: On the translocation of masses. Compt. Rend. Akad. Sei **7**, 199–201 (1942)