Non-convex min-max Optimization

Shashwat Gupta, *IIT Kanpur, India*,
Sebastian Breguel, *EPFL Lausanne, Switzerland*
Martin Jaggi, *EPFL Lausanne, Switzerland*
Nicolas Flammarion, *EPFL Lausanne, Switzerland*

***Abstract*—In this short study, we aim to gain deeper insights to Keswani's algorithm [1] for sequential minimax optimisation, by comparing the behaviour with 2 other algorithms : Gradient Descenet Ascent (GDA) and Optimistic Mirror Descent (OMD).**

I. Introduction

We consider differentiable sequential games with two players: a leader who can commit to an action, and a follower who responds after observing the leader's action. Particularly, we focus on the zero-sum case of this problem which is also known as minimax optimization, i.e.,

$$\min_{x\in\mathbb{R}^n}\max_{y\in\mathbb{R}^m} f(x,y)$$

Unlike simultaneous games, many practical machine learning algorithms, including generative adversarial networks (GANs) [2] [3] , adversarial training [4] and primal-dual reinforcement learning [5], explicitly specify the order of moves between players and the order of which player acts first is crucial for the problem. In particular, min-max optimisation is curcial for GANs [2], statistics, online learning [6], deep learning, and distributed computing [7]. Therefore, the classical notion of local Nash equilibrium from simultaneous games may not be a proper definition of local optima for sequential games since minimax is in general not equal to maximin. Instead, we consider the notion of local minimax [8] which takes into account the sequential structure of minimax optimization.

II. Models and Methods

The vanilla algorithm for solving sequential minimax optimization is gradient descent-ascent (GDA), where both players take a gradient update simultaneously. However, GDA is known to suffer from two drawbacks.

1) It has undesirable convergence properties: it fails to converge to some local minimax and can converge to fixed points that are not local minimax [9] [10]
2) GDA exhibits strong rotation around fixed points, which requires using very small learning rates[11] [12] to converge.

Recently, there has been a deep interest in minmax problems, due to [9] and other subsequent works. Jin et al. [8] actually provides great insights to the work.

For the project, we try to implement the algorithm by Keswani et. al. [1] , and try to deduce various aspects of it.

III. Insight into Keswani's algorithm

The algorithm essentially makes response function : $\max_{y\in\mathbb{R}^m} f(.,y)$ tractable by selecting y-updates (maxplayer) in greedy manner by restricting selection of updated (x,y) to points along sets P(x,y) (which is defined as set of endpoints of paths such that f(x,.) is non-decreasing). There are 2 new things that this algorithm does to make computation feasible:

1) Replace $P(x,y)$ with $P_\varepsilon(x,y)$ (endpoints of paths along which f(x,.) increases at some rate $\varepsilon > 0$ (which makes updates to y by any 'greedy' algorithm (as Algorithm 2) feasible)
2) Introduce a 'soft' probabilistic condition to account for discountinous functions.

Algorithm Keswani algorithm for min-max optimization

input: Stochastic zeroth-order oracle F for bounded loss function $f:\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}$ with L-Lipschitz gradient, stochastic gradient oracle G_y with mean $\nabla_y f$, Initial point (x_0,y_0)
input: A distribution $Q_{x,y}$, and an oracle for sampling from this distribution. Error parameters $\varepsilon,\delta>0$
hyperparameters: $\eta>0$ (learning rate), $r_{\max}$ (maximum number of rejections); τ_1 (for annealing);

Set $i\leftarrow 0$, $r\leftarrow 0$, $\varepsilon_0=\frac{\varepsilon}{2}$, $f_{\text{old}}\leftarrow\infty$
while $r\le r_{\max}$ **do**
  Sample Δ_i from the distribution Q_{x_i,y_i}
  Set $X_{i+1}\leftarrow x_i+\Delta_i$ {*min-player's proposed update*}
  Run Algorithm 2 with inputs $\mathrm{x}\leftarrow X_{i+1}$, $\mathrm{y}_0\leftarrow y_i$, and $\varepsilon'\leftarrow\varepsilon_i\times(1-2\eta L)^{-1}$ {*max-player's update*}
  Set $\mathcal{Y}_{i+1}\leftarrow \mathrm{y}_{\text{stationary}}$ to be the output of Algo 2.
  Set $f_{\text{new}}\leftarrow F(X_{i+1},\mathcal{Y}_{i+1})$ {*Compute the new loss*}
  Set $\mathsf{Accept}_i\leftarrow\mathsf{True}$.
  if $f_{\text{new}}>f_{\text{old}}-\frac{\delta}{4}$, **then**
    Set $\mathsf{Accept}_i\leftarrow\mathsf{False}$ with probability $\max(0,1-e^{-\frac{i}{\tau_1}})$ {*Decide to accept or reject*}
  if $\mathsf{Accept}_i=\mathsf{True}$ **then**
    Set $x_{i+1}\leftarrow X_{i+1}$, $y_{i+1}\leftarrow\mathcal{Y}_{i+1}$ {*accept the proposed x and y updates*}
    Set $f_{\text{old}}\leftarrow f_{\text{new}}$, $r\leftarrow 0$, $\varepsilon_{i+1}\leftarrow\varepsilon_i\times(1-2\eta L)^{-2}$
  else
    Set $x_{i+1}\leftarrow x_i$, $y_{i+1}\leftarrow y_i$, $r\leftarrow r+1$, $\varepsilon_{i+1}\leftarrow\varepsilon_i$ {*Reject the proposed updates*}
  Set $i\leftarrow i+1$
return $(x^\star,y^\star)\leftarrow(x_i,y_i)$

Fig. 1: Keswani's Algorithm adapted from the original paper (Algorithm 2 is the optimisation algorithm to compute max-player updates.)

IV. Experiment

In this experiment, we compare the convergence behaviour of various functions using our MATLAB Code with three different optimizers, a) GDA, B) OMD c) Keswani's Algorithm [1]. The functions chosen are the typical examples from recent ICML,ICLR papers on min-max optimisation, with some specific properties associated with them.

We choose the following functions for this experiment:

1) $F_1(x,y)=-3x^2-y^2+4xy$[1][9] : specified in [1] and has stationary point at (0,0), GDA, OMD and Extra-gradient (EG) do not converge on it as shown in [9]
2) $F_2(x,y)=3x^2+y^2+4xy$[1][9] :

3) $F_3(x,y) = (4x^2 - (y - 3x + 0.05x^3)^2 - 0.1y^4)e^{-0.01(x^2+y^2)}$ [1] : specified in [1] and has stationary point at (0,0), GDA,OMD and EG do not converge on it as shown in [9]
4) $F_4(x,y) = xy - \frac{1}{3}y^3$[13]
5) $F_5(x,y) = x^2 + 3\sin(x)\sin(y) - 4y^2 - 10\sin(y)$[14]
6) $F_6(x,y) = 10xy$[15]
7) $F_7(x,y) = 0.5x^2 + 10xy + 0.5y^2$[15]
8) $F_8(x,y) = 10xy - y^2$[15]
9) $F_9(x,y) = sin(x+y)$[8] : No strategic Nash equilibrium (global or local) chat
10) $F_{10}(x,y) = 0.2xy - cos(y)$[8] : Global minmax can neither be local minimax nor a stationary point , Twice differentiable s.t. the point is Evtushenko minimax (defined in [6]) for W_1, but not for W_2; $W_1 = [-1,1] \times [-2\pi, 0]$ and $W_2 = [-1,1] \times [-2\pi, 2\pi]$
11) $F_{11}(x,y) = -0.03x^2 + 0.2xy - cos(y)$[8] : the function has Evshenko minimax (defined in [8]) optima in [-1,1] × [-2π,2π] at (0,$-\pi$) which is non-stationary point and does not satisfy condition at Jim et al.

V. Setup and Results

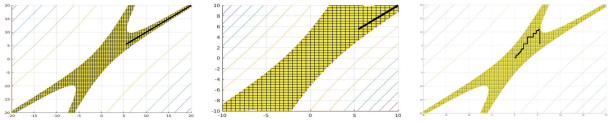
In this experiment, we compare the result and time taken by keswani's algorithm VS Standard algorithms (GDA and OMD), and try to comment on the types of functions for which the Keswani's algorithm is suitable.

The code is uploaded on the github link. There are 3 MATLAB files : GDA.m, OMD.m and OurAlgorithm.m. The files contain all the code that needs to be run. The relevant functions are implemented at the end of the files. To run Function 1, one can uncomment the section : expression under $\%\%\%F1$ in function z=value(x,y), expression under $\%\%\%nabla\_yF1$ in function g=xGrad(x,y) and under $\%\%\%nable\_yF1$ under function g=yGrad(x,y)

We use eta=0.05 and sigma=0.5 for keswani's algorithm. Delta = 0.03 and epsilon=0.001. Max-reject = 20. T=20000 for keswani algorithm and 401 (usually) for GDA and OMD. The usual iterations are 400, however, if we see some trend (eg. cycling, mode-collapse) forming, we increase iterations to complete that trend.

We share our code here: https://github.com/ShashwatGupta2001/CS439-Optimisation4ML-2023.git

A. Function 1:

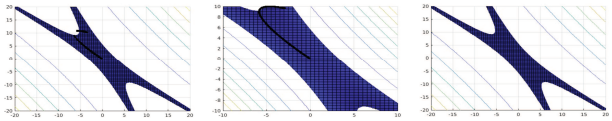


(a) GDA (b) OMD (c) Keswani's Algorithm

Fig. 2: $F_1(x,y) = -3x^2 - y^2 + 4xy$

Result: a) GDA and b) OMD : how was showed in [9] not converges. Instead of that, diverges. c) Keswani's Algorithm: Convergence to the point $(0,0)$.

B. Function 2:

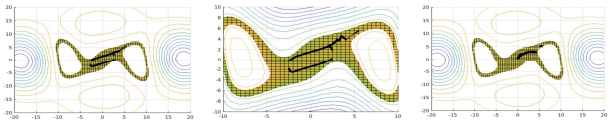


(a) GDA (b) OMD (c) OURs

Fig. 3: $F_2(x,y) = 3x^2 + y^2 + 4xy$

Result: a) GDA : Don't converge to the global min-max. $(0,0)$. b) OMD: Don't converge to the global min-max. $(0,0)$. c) Keswani's Algorithm: Converge to $(0,0)$. that is the global min-max

C. Function 3:

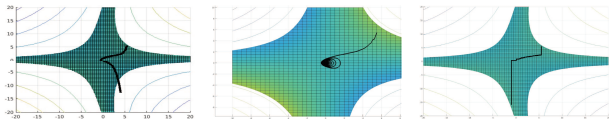


(a) GDA (b) OMD (c) Keswani's Algorithm

Fig. 4: $F_3(x,y) = (4x^2 - (y - 3x + 0.05x^3)^2 - 0.1y^4)e^{-0.01(x^2+y^2)}$

Result: a) GDA and b) OMD do not converge but cycld around $(0,0)$ c) Keswani's Algorithm: Converge to $(0,0)$. that is the global min-max.

D. Function 4:

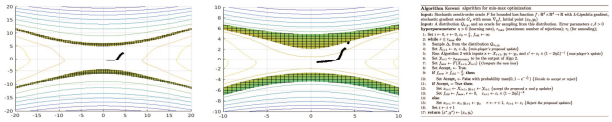


(a) GDA (b) OMD (c) OURs

Fig. 5: $F_4(x,y) = xy - \frac{1}{3}y^3$

Results: a) GDA: starts converging to a point but ends up diverging. b) OMD: Converge to $(0,0)$. c) Keswani's Algorithm: starts converging to the saddle point and ends in divergence. The behaviour of a. and b. seems just like motion of planets, prompting to explore the mathematical similarity between vectors here to vectors that arise in gravitational field.

E. Function 5:



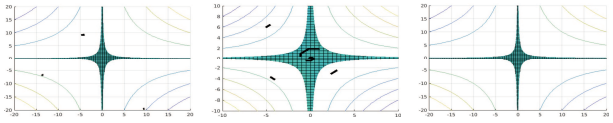
(a) GDA (b) OMD (c) Keswani's Algorithm

Fig. 6: $F_5(x,y) = x^2 + 3\sin(x)\sin(y) - 4y^2 - 10\sin(y)$

Result: GDA Converges to (x,y)=(0.7429,-0.73747) in 0.607s. OMD also converges to (0.7429,-0.7375). Both the

points are similar. But they differ from Keswani's algorithm, which converges to (x,y) = (0.6134,-0.7539) in just 353 iterations.

F. Function 6:

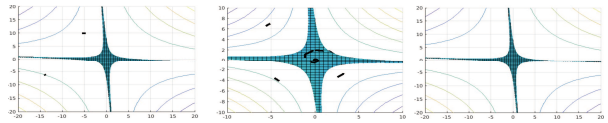


(a) GDA (b) OMD (c) Keswani's Algorithm

Fig. 7: $F_6(x,y) = 10xy$

Result: Surprisingly, for the function (similar to 8), the GDA and Keswani diverge, however OMD converges to (0,0) also.

G. Function 7:

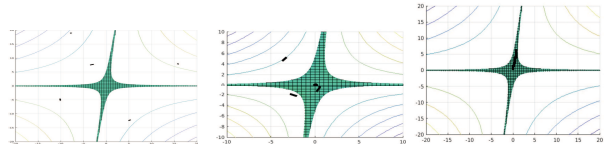


(a) GDA (b) OMD (c) Keswani's Algorithm

Fig. 8: $F_7(x,y) = 0,5x^2 + 10xy + 0.5y^2$

Result: Surprisingly, for the function (similar to 6 and 8), the GDA and Keswani diverge, however OMD converges to (0,0) also. (result similar to 6)

H. Function 8:

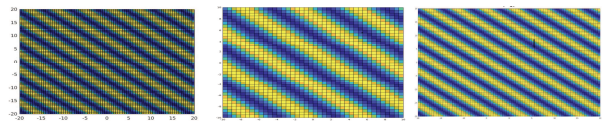


(a) GDA (b) OMD (c) Keswani's Algorithm

Fig. 9: $F_8(x,y) = 10xy - y^2$

Result: The GDA Algorithm traverses to infinity (x,y) = (4.824591e+12,-8.489846e+12) in about 0.6 s. The OMD algorithm however, converges to (0,0) ((x,y)=(2.073641e-60,-3.133129e-60) to be exact) in 0.16 s. Keswani's algorithm also converges to Origin, in 0.3706s (with t=51993 and i=66).

I. Function 9:

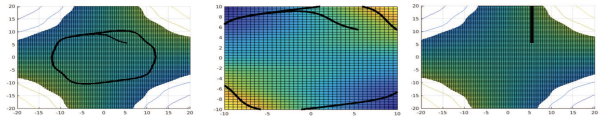


(a) GDA (b) OMD (c) Keswani's Algorithm

Fig. 10: $F_9(x,y) = \sin(x+y)$

Result: The GDA Algorithm does not traverse far and essentially remains confined to the point. This is mainly because the problem can not be solved using strategy based Nash equilibrium. OMD also does not travel far as well. Keswani's algorithm seems to travel some distance and reach (x,y) = (5.500,8.636) which is a local max point (with i=21) and stays there.

J. Function 10:

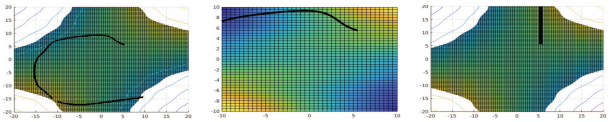


(a) GDA (b) OMD (c) Keswani's Algorithm

Fig. 11: $F_{10}(x,y) = 0.2xy - \cos(y)$

Result: a) GDA : Cycles through for this function.Thus, we do not get any sensible x-value for the convergence. b) OMD: Cycles around the optimal point. Thus no convergence c) Keswani's Algorithm: Diverges to infinity straightaway, conveying that there is no such minimax point (no i is accepted and all iterations were run). For initialisation at (0,0), the algorithm increases iterations (i), but does not seem to converge, so stays close to initialised point.

K. Function 11:



(a) GDA (b) OMD (c) Keswani's Algorithm

Fig. 12: $F_{11}(x,y) = -0.03x^2 + 0.2xy - \cos(y)$

Result: GDA cycles through for this function.Though the function is similar to 10, we find that the function initially loops for a half-cycle and then diverges. The reported x,y values are (-2.390e+03,3.406e+03). OMD, just like GDA, curves then diverges. The reported x,y values are (-28.273,-35.884). Keswani's Algorithm: Just like 10, the algorithm moves around the initialisation point. A different initialisation eg. (0,0) does not do any better.

VI. Conclusion and Future work

This research builds upon Keswani's work and opens new avenues for exploration, from the work of Keswani and our conclusions.

1) Exploring Stricter Bounds: We used the Keswani's bound.However as mentioned by them, we could explore proving linear bounds for more efficiency. .
2) Incorporating Different Function Categories: Our study leveraged a specific category of functions to generate insights. A next step, to broaden this scope could be introducing other categories of functions.
3) Comparison of Optimizers' Performance: Keswani's algorithm employed a specific optimizer (Algorithm 2)

SGD (Stochastic Gradient Descent) for its computations. Future research could focus on comparing the performance and effects of different optimizers on the algorithm's performance. Our repo has a jupyter notebook to compare the effect of different optimizers on GDA and Keswani's algorithm.

REFERENCES

[1] V. Keswani, O. Mangoubi, S. Sachdeva, and N. K. Vishnoi, "A convergent and dimension-independent first-order algorithm for min-max optimization," *arXiv preprint arXiv:2006.12376*, 2020.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[3] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," pp. 214–223, 2017.

[4] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[5] W. S. Cho and M. Wang, "Deep primal-dual reinforcement learning: Accelerating actor-critic using bellman duality," *arXiv preprint arXiv:1712.02467*, 2017.

[6] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[7] J. Shamma, *Cooperative Control of Distributed Multi-Agent Systems*. Wiley & Sons, Incorporated, John, 2008.

[8] C. Jin, P. Netrapalli, and M. Jordan, "What is local optimality in nonconvex-nonconcave minimax optimization?" pp. 4880–4889, 2020.

[9] Y. Wang, G. Zhang, and J. Ba, "On solving minimax optimization locally: A follow-the-ridge approach," *arXiv preprint arXiv:1910.07512*, 2019.

[10] C. Daskalakis and I. Panageas, "The limit points of (optimistic) gradient descent in min-max optimization," *Advances in neural information processing systems*, vol. 31, 2018.

[11] L. Mescheder, S. Nowozin, and A. Geiger, "The numerics of gans," *Advances in neural information processing systems*, vol. 30, 2017.

[12] D. Balduzzi, S. Racaniere, J. Martens, J. Foerster, K. Tuyls, and T. Graepel, "The mechanics of n-player differentiable games," pp. 354–363, 2018.

[13] D. M. Ostrovskii, B. Barazandeh, and M. Razaviyayn, "Nonconvex-nonconcave min-max optimization with a small maximization domain," *arXiv preprint arXiv:2110.03950*, 2021.

[14] J. Yang, N. Kiyavash, and N. He, "Global convergence and variance reduction for a class of nonconvex-nonconcave minimax problems," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1153–1165, 2020.

[15] G. Zhang, Y. Wang, L. Lessard, and R. B. Grosse, "Near-optimal local convergence of alternating gradient descent-ascent for minimax optimization," pp. 7659–7679, 2022.