

A Decentralized Proximal Point-type Method for Non-convex Non-concave Saddle Point Problems

Weijie Liu, Aryan Mokhtari, Asu Ozdaglar, Sarath Pattathil, Zebang Shen, Nenggan Zheng

Problem Formulation and Assumptions

- We are interested in solving the following saddle point problem:

$$\min_{\mathbf{x} \in \mathcal{X}} \max_{\mathbf{y} \in \mathcal{Y}} f(\mathbf{x}, \mathbf{y})$$

- Used to formulate GANs, Adversarial training, Robust Optimization
- Many papers study the setting where the objective is convex-concave. But most practical applications involve nonconvex functions.
- Many applications involve a formulation where the objective function can be written as a sum of other functions. Each of these functions are evaluated using data from a node and we would like to solve the optimization problem using minimal communication between the nodes.
- The objective in this paper is to find the saddle point of the function $f(x, y)$ where $f = \sum_{i=1}^n f_i(x, y)$ is the sum of functions f_i where we assume that each of these functions are assigned to a node i

Definition Consider a function $\phi : \mathbb{R}^p \rightarrow \mathbb{R}$. The function ϕ is

(a) convex over \mathcal{X} if for any $\mathbf{x}, \hat{\mathbf{x}} \in \mathcal{X}$ we have $\phi(\hat{\mathbf{x}}) \geq \phi(\mathbf{x}) + \langle \nabla \phi(\mathbf{x}), \hat{\mathbf{x}} - \mathbf{x} \rangle$.

(b) μ -strongly convex over \mathcal{X} if for any $\mathbf{x}, \hat{\mathbf{x}} \in \mathcal{X}$ we have $\phi(\hat{\mathbf{x}}) \geq \phi(\mathbf{x}) + \langle \nabla \phi(\mathbf{x}), \hat{\mathbf{x}} - \mathbf{x} \rangle + \frac{\mu}{2} \|\hat{\mathbf{x}} - \mathbf{x}\|^2$.

(c) ρ -weakly convex over \mathcal{X} if for any $\mathbf{x}, \hat{\mathbf{x}} \in \mathcal{X}$ we have if $\phi(\hat{\mathbf{x}}) \geq \phi(\mathbf{x}) + \langle \nabla \phi(\mathbf{x}), \hat{\mathbf{x}} - \mathbf{x} \rangle - \frac{\rho}{2} \|\hat{\mathbf{x}} - \mathbf{x}\|^2$.

Further, a function $\phi(\mathbf{x})$ is concave, μ -strongly concave, or ρ -weakly concave, if $-\phi(\mathbf{x})$ is convex, μ -strongly convex, or ρ -weakly convex, respectively.

Now, we state the assumptions on the objective function

Assumption 1 The objective function $f(\mathbf{x}, \mathbf{y})$ is ρ -weakly convex with respect to \mathbf{x} and ρ -weakly concave with respect to \mathbf{y} for all $\mathbf{x} \in \mathcal{X}$ and $\mathbf{y} \in \mathcal{Y}$.

Assumption 2 The sets \mathcal{X} and \mathcal{Y} are convex, closed and bounded.

Under these assumptions, we show convergence to a neighborhood around a first order stationary point. Under further assumptions (MVI), we show exact convergence

Assumption 3 (Minty Variational Inequality [MVI]): There exists a point \mathbf{z}^* such that for every local operator $\mathcal{B}_n, \mathcal{B}_n(\mathbf{z})^\top (\mathbf{z} - \mathbf{z}^*) \geq 0$ for any $\mathbf{z} \in \mathcal{Z}$.

This assumption includes the class of convex-concave problems. It also includes some other classes of functions (operators) such as Pseudo Monotone operators

Motivation and Overview

- Saddle Point problems have applications in several Machine Learning and Robust control problems
- In this work we propose the first algorithm to solve the distributed version of the nonconvex-nonconcave minimax problem.
- General Problem is NP-Hard.
- We first show that our proposed algorithm converges to a neighborhood of the solution.
- Under the stronger MVI assumption, we show exact convergence to the solution
- Finally, we have numerical experiments to show the superior performance of our algorithm.

Algorithm DPPSP at node n

Input: initial iterate \mathbf{z}_n^0 , step size α , weights w_{nm} for $m \in \mathcal{N}_n$;

- for** $t = 0, \dots, T - 1$ **do**
- Exchange variable \mathbf{z}_n^t with neighboring nodes $m \in \mathcal{N}_n$;
- if** $t = 0$ **then**
- $\mathbf{z}^{t+1} = (\mathbf{I} + \alpha(\mathcal{B}_n + \mathcal{R}_n))^{-1} (\sum_{m \in \mathcal{N}_n} (2w_{nm} - 1)\mathbf{z}_m^t)$;
- else**
- $\mathbf{z}_n^{t+1} = (\mathbf{I} + \alpha(\mathcal{B}_n + \mathcal{R}_n))^{-1} (\sum_{m \in \mathcal{N}_n} w_{nm}(2\mathbf{z}_m^t - \mathbf{z}_m^{t-1}) + \alpha[\mathcal{B}(\mathbf{z}_n^t) + \mathcal{R}(\mathbf{z}_n^t)])$;
- end if**
- end for**

Theorem Consider the DPPSP method outlined in the Algorithm. Suppose the conditions in Assumption 1-2 are satisfied and the stepsize is chosen such that $\alpha \leq 1/(2\rho)$. If we run DPPSP for T iterations and choose one of the iterates s uniformly at random from the time indices $1, \dots, T$, then

$$\mathbb{E}_s \left[\left\| \sum_{n=1}^N \mathcal{B}_n(\mathbf{z}_n^s) + \mathcal{R}_n(\mathbf{z}_n^s) \right\| \right] \leq \frac{1}{\alpha} \sqrt{\frac{N}{\lambda_{\min}(\tilde{\mathbf{W}})}} \left(\frac{\|\phi^0 - \phi^*\|_{\mathbf{M}}}{\sqrt{T}} + \sqrt{2\alpha\rho ND} \right),$$

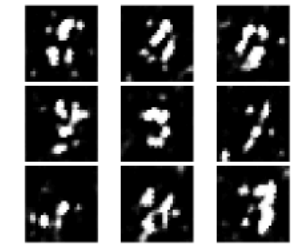



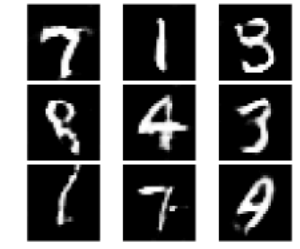
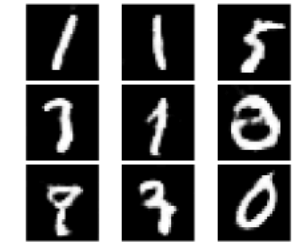


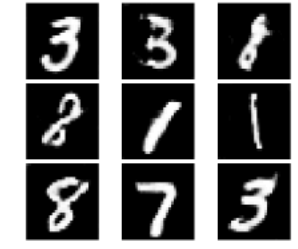



$$\mathbb{E}_s [\|\mathbf{U}\mathbf{z}^{s+1}\|] \leq \frac{\|\phi^0 - \phi^*\|_{\mathbf{M}}}{\sqrt{T}} + \sqrt{2\alpha\rho ND},$$

where $\|\phi^0 - \phi^*\|_{\mathbf{M}} = \|\mathbf{z}_0 - \mathbf{z}^*\|_{\tilde{\mathbf{W}}} + \|\mathbf{U}\mathbf{z}_0 - \mathbf{q}^*\|$

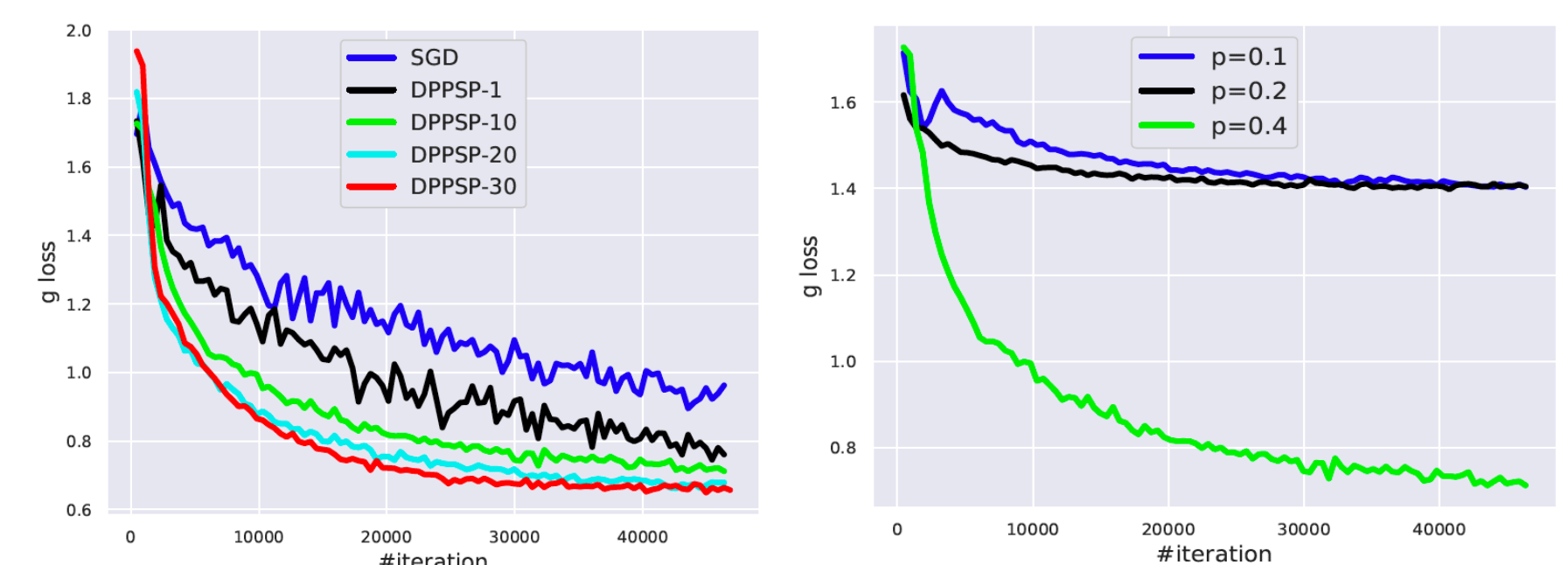
Theorem Consider the DPPSP method outlined in the Algorithm. Suppose Assumption 1-3 hold and the stepsize is chosen such that $\alpha = 1/(2\rho)$. If we run the DPPSP algorithm for T iterations and choose one of the iterates s uniformly at random from the time indices $1, \dots, T$ then we have

$$\mathbb{E}_s \left[\left\| \sum_{n=1}^N \mathcal{B}_n(\mathbf{z}_n^{s+1}) + \mathcal{R}_n(\mathbf{z}_n^{s+1}) \right\| \right] \leq \frac{ND}{\alpha\sqrt{T}}, \quad \mathbb{E}_s [\|\mathbf{U}\mathbf{z}^s\|] \leq \frac{\sqrt{ND}}{\sqrt{T}}.$$

Numerical Results

iterations	2340	9360	28080
SGD			
DPPSP-1			
DPPSP-10			
DPPSP-20			

Images produced by these methods after 2340, 9360, and 28080 iterations. We observe that after 9360 iterations, DPPSP-20 and DPPSP-30 already have generated reasonable images. Also, images of DPPSP-20 have better quality than the ones generated by DPPSP-10. Generators trained by SGD and DPPSP-1 output unsatisfactory samples even after 28080 iterations. According to these results, increasing the number of processing units not only leads to a smaller loss for the generator but also produces images with higher quality.



We use SGD as baseline to validate the performance of the proposed DPPSP method. We show the performances of DPPSP with varying number of nodes and different graph connectivity on the MNIST dataset. Single-node DPPSP method outperforms SGD. The advantage of DPPSP is more significant when we have more processing nodes. Note that the number of samples used per iteration are identical for all the considered settings.