Accelerated Gradient-Free Method for Heavily Constrained Nonconvex Optimization

Anonymous authors
Paper under double-blind review

Abstract

Zeroth-order (ZO) method has been shown to be a powerful method for solving the optimization problem where explicit expression of the gradients is difficult or infeasible to obtain. Recently, due to the practical value of the constrained problems, a lot of ZO Frank-Wolfe or projected ZO methods have been proposed. However, in many applications, we may have a very large number of nonconvex white/black-box constraints, which makes the existing zeroth-order methods extremely inefficient (or even not working) since they need to inquire function value of all the constraints and project the solution to the complicated feasible set. In this paper, to solve the nonconvex problem with a large number of white/black-box constraints, we proposed a doubly stochastic zeroth-order gradient method (DSZOG). Specifically, we reformulate the problem by using the penalty method with distribution probability and sample a mini-batch of constraints to calculate the stochastic zeroth/first-order gradient of the penalty function to update the parameters and distribution, alternately. To further speed up our method, we propose an accelerated doubly stochastic zeroth-order gradient method (ADSZOG) by using the exponential moving average method and adaptive stepsize. Theoretically, we prove DSZOG and ADSZOG can converge to the ϵ-stationary point of the constrained problem. We also compare the performances of our method with several ZO methods in two applications, and the experimental results demonstrate the superiority of our method in terms of training time and accuracy.

1 Introduction

Zeroth-order (gradient-free) method has been shown to be a powerful method for solving the optimization problem where explicit expression of the gradients are difficult or infeasible to obtain, such as bandit feedback analysis Agarwal et al. (2010), reinforcement learning Choromanski et al. (2018), and adversarial attacks on black-box deep neural networks Chen et al. (2017); Liu et al. (2018a). Zeroth-order (ZO) methods only use the function values to approximate the full gradient or stochastic gradient, and then the gradient descent can be used. Due to the friendly of approximating the gradient and scalability to large scale problems, more and more zeroth-order gradient algorithms have been proposed and achieved great success, such as Ghadimi & Lan (2013); Wang et al. (2018); Gu et al. (2016); Liu et al. (2018a); Huang et al. (2020a).

Recently, constrained optimizations have become increasingly relevant to the machine learning community. Due to several motivating applications, the study of the zeroth-order methods in constrained optimization has gained great attention. Specifically, ZOSCGDBalasubramanian & Ghadimi (2018) uses the zeroth-order method to approximate the unbiased stochastic gradient of the objective, and then use the Frank-Wolfe framework to update the model parameters. Based on ZOSCGD, Gao & Huang (2020) use the variance reduction technique Fang et al. (2018); Nguyen et al. (2017) to obtain a better convergence performance. In addition, Huang et al. (2020b) use the variance reduction technique and momentum acceleration technique to further speed up the ZO Frank-Wolfe method. ZOSPGD Liu et al. (2018b) uses the ZO method to update the parameters and then projects the solution onto the feasible subset. ZOADAMM$^+$ Liu et al. (2020) uses the adaptive momentum method to accelerate the ZOSPGD. We have summarized several representative zeroth-order methods for constrained optimization in Table 1.

Table 1: Representative zeroth order methods for constrained optimization problems, where N/C means nonconvex/convex, W/B means white/black-box function, and the last column shows the size of the constraints.

Framework	Algorthm	Reference	Objective	Constraints	Size
Frank-Wolfe	ZOSCGD	Balasubramanian & Ghadimi (2018)	N/C	C W	Small
	FZFW	Gao & Huang (2020)	N/C	C W	Small
	FZCGS				
	FCGS				
	Acc-SZOFW	Huang et al. (2020b)	N/C	C W	Small
	Acc-SZOFW*				
Projected	ZOPSGD	Liu et al. (2018b)	N/C	C W	Small
	ZOADAMM$^+$	Liu et al. (2020)	N/C	C W	Small
Penalty	DSZOG ADSZOG	Ours	N/C	N/C W/B	Large

However, all these methods only focus on the simple constrained problem and are not scalable for the problems with a large number of constraints. Specifically, on the one hand, they all need to evaluate the values of all the constraints in each iteration. On the other hand, the projected gradient methods and the Frank-Wolfe methods need to solve a subproblem in each iteration. These makes the existing methods time-consuming to find a point satisfying all the constraints. What's worse, all these methods need the constraints to be convex white-box functions. However, in many real-world applications, the constraints could be nonconvex or black-box functions, which makes the existing method extremely inefficient or even not working. Therefore, how to effectively solve the nonconvex constrained problem with a large number of nonconvex/convex white/black-box constraints, which is denoted as heavily constrained problem, by using the ZO method is still an open problem.

In this paper, to solve the heavily constrained nonconvex optimization effectively and efficiently, we propose two new ZO algorithms called doubly stochastic zeroth-order gradient method (DSZOG) and accelerated doubly stochastic zeroth-order gradient method (ADSZOG). Specifically, we give a probability distribution over all the constraints and rewrite the original problem as a nonconvex-strongly-concave minimax problem Lin et al. (2020); Wang et al. (2020); Huang et al. (2020a); Guo et al. (2021) with respect to the model parameter and probability distribution by using the penalty method. We first sample a mini-batch of training points uniformly and a mini-batch of constraints according to the distribution to calculate the zeroth-order gradient of the penalty function w.r.t model parameters and then sample a mini-batch of constraints uniformly to calculate the stochastic gradient of penalty function w.r.t the probability distribution. Then, gradient descent and projected gradient ascent can be used to update model parameters and probability distribution. In addition, to further speed up training, we propose a new accelerated doubly stochastic zeroth-order gradient method by using the exponential moving average (EMA) method and adaptive stepsize Guo et al. (2021); Huang et al. (2020a), which benefits our method from the variance reduction and adaptive convergence. Theoretically, we prove DSZOG and ADSZOG can converge to the ϵ-stationary point of the constrained problem. We also compare the performances of our method with several ZO methods in two applications, and the experimental results demonstrate the superiority of our method in terms of training time and accuracy.

Contributions. We summarized the main contributions of this paper as follows:

1. We propose a doubly stochastic zeroth-order gradient method to solve the heavily constrained nonconvex problem. By introducing a stochastic layer into the constraints, our method is scalable and efficient for the heavily constrained nonconvex problem.
2. We also proposed an accelerated doubly stochastic zeroth-order gradient method to solve the heavily constrained nonconvex problem. By using the exponential moving average method and adaptive stepsize, it enjoys the benefits of variance reduction and adaptive convergence.
3. We prove DSZOG and ADSZOG can converge to the ϵ-stationary point of the constrained problem. Experimental results also demonstrate the superiority of our methods in terms of accuracy and training time.

2 Proposed Method

2.1 Problem Setting

In this paper, we consider the following nonconvex constrained problem,

$$\min_{\boldsymbol{w}} \; f_0(\boldsymbol{w}) := \frac{1}{n}\sum_{i=1}^{n} \ell_i(\boldsymbol{w}) \; s.t. \; f_j(\boldsymbol{w}) \leq 0, \; j = 1, \cdots, m, \tag{1}$$

where $\boldsymbol{w} \in \mathbb{R}^d$ is the optimization variable, $\{\ell_i(\boldsymbol{w})\}_{i=1}^n$ are n component functions. In addition, $f_0 : \mathbb{R}^d \mapsto \mathbb{R}$ is a nonconvex and black-box function. $f_j : \mathbb{R}^d \mapsto \mathbb{R}$, $(j = 1, \cdots, m)$, is nonconvex/convex and white/black-box function. Such a problem is denoted as heavily constrained problem Cotter et al. (2016).

2.2 Reformulate the constrained problem

To solve the constrained problem, the penalty method is one of the main approaches. Following this method, we reformulate the constrained optimization problem as the following minimax problem over a probability distribution Clarkson et al. (2012); Cotter et al. (2016)

$$\min_{\boldsymbol{w}} \max_{\boldsymbol{p} \in \Delta^m} \mathcal{L}(\boldsymbol{w}, \boldsymbol{p}) = f_0(\boldsymbol{w}) + \beta\varphi(\boldsymbol{w}, \boldsymbol{p}) - \frac{\lambda}{2}\|\boldsymbol{p}\|_2^2 \tag{2}$$

where $\beta > 0$, $\lambda > 0$, $\varphi(\boldsymbol{w}, \boldsymbol{p}) = \sum_{j=1}^m p_j \phi_j(\boldsymbol{w})$, $\phi_j(\boldsymbol{w}) = (\max\{f_j(\boldsymbol{w}), 0\})^2$ is the penalty function on f_j, $\Delta^m := \{\boldsymbol{p} | \sum_{j=1}^d p_j = 1, 0 \leq p_j \leq 1, \forall j \in [d]\}$ is the m-dimensional simplex and $\boldsymbol{p} = [p_1, \cdots, p_m] \in \Delta^m$. Note different the formulation in Clarkson et al. (2012); Cotter et al. (2016), there is an additional term $-\frac{1}{2}\|\boldsymbol{p}\|_2^2$ which is used to ensure $\mathcal{L}$ to be strongly concave on $\boldsymbol{p}$.

2.3 Doubly zeroth-order stochastic gradient method

Since we can only obtain the values of the objective and constraints, we use the zeroth-order gradient method to solve this minimax problem 2. Obviously, calculating the zeroth-order gradient of $\mathcal{L}$ needs to inquire the function values of all the constraints and ℓ_i, which has a very high time complexity if m and n are very large.

To solve this problem, we use the stochastic manner. Specifically, since the minimax problem 2 contains two finite sums, i.e., $f_0(\boldsymbol{w}) = 1/n \sum_{i=1}^n \ell_i(\boldsymbol{w})$ and $\varphi(\boldsymbol{w}, \boldsymbol{p}) = \sum_{j=1}^m p_j \phi_j(\boldsymbol{w})$, we can calculate their stochastic zeroth-order gradient, respectively, and then obtain the stochastic zeroth-order gradient of $\mathcal{L}$ w.r.t. $\boldsymbol{w}$. We first calculate the stochastic zeroth-order gradient of f_0 and $\varphi(\boldsymbol{w}, \boldsymbol{p})$ as follows,

$$G_\mu^f(\boldsymbol{w}_t, \ell_i, \boldsymbol{u}) = \frac{\ell_i(\boldsymbol{w}_t + \mu\boldsymbol{u}) - \ell_i(\boldsymbol{w}_t)}{\mu}\boldsymbol{u}, \quad G_\mu^\varphi(\boldsymbol{w}_t, \boldsymbol{p}, f_j, \boldsymbol{u}) = \frac{\phi_j(\boldsymbol{w}_t + \mu\boldsymbol{u}) - \phi_j(\boldsymbol{w}_t)}{\mu}\boldsymbol{u},$$

by sampling a ℓ_i uniformly, and a f_j according to $\boldsymbol{p}$, where $\mu > 0$ and $\boldsymbol{u} \sim \mathcal{N}(0, \mathbf{1}_d)$. Note here we sample the constraint according to the distribution $\boldsymbol{p}$, which makes our method can find the most-violated constraint in each iteration Cotter et al. (2016). Then, combining these two terms, we can obtain the stochastic zeroth-order gradient of $\mathcal{L}$ w.r.t. $\boldsymbol{w}$ as follows,

$$G_\mu^{\mathcal{L}}(\boldsymbol{w}_t, \boldsymbol{p}_t, \ell_i, f_j, \boldsymbol{u}) = G_\mu^f(\boldsymbol{w}_t, \ell_i, \boldsymbol{u}) + \beta G_\mu^\varphi(\boldsymbol{w}_t, \boldsymbol{p}_t, f_j, \boldsymbol{u}).$$

To reduce the variance, we can sample a batch of ℓ_i, f_j and $\boldsymbol{u}_k$ to calculate the zeroth-order gradient. Given $q > 0$, $\mathcal{M}_1 \subseteq [n]$ and $\mathcal{M}_2 \sim \boldsymbol{p} \subseteq [m]$, we have

$$G_\mu^{\mathcal{L}}(\boldsymbol{w}_t, \boldsymbol{p}_t, \ell_{\mathcal{M}_1}, f_{\mathcal{M}_2}, \boldsymbol{u}_{[q]}) = \frac{1}{q|\mathcal{M}_1|}\sum_{i \in \mathcal{M}_1}\sum_{k=1}^{q} G_\mu^f(\boldsymbol{w}_t, \ell_i, \boldsymbol{u}_k) + \frac{\beta}{q|\mathcal{M}_2|}\sum_{j \in \mathcal{M}_2}\sum_{k=1}^{q} G_\mu^\varphi(\boldsymbol{w}_t, \boldsymbol{p}_t, f_j, \boldsymbol{u}_k)$$

Then, we can use $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_{\boldsymbol{w}} G_\mu^{\mathcal{L}}(\boldsymbol{w}_t, \boldsymbol{p}_t, \ell_{\mathcal{M}_1}, f_{\mathcal{M}_2}, \boldsymbol{u}_{[q]})$ to update $\boldsymbol{w}$.

Then, we also use stochastic gradient to update the distribution $\boldsymbol{p}$. In each iteration, we randomly sample a constraint $f_j(\boldsymbol{w})$ to calculate the stochastic gradient of $\mathcal{L}$ w.r.t. $\boldsymbol{p}$ by using

$$H(\boldsymbol{w}_t, \boldsymbol{p}_t, f_j) = \beta m \boldsymbol{e}_j \phi_j(\boldsymbol{w}_t) - \lambda \boldsymbol{p}_t,$$

where $\boldsymbol{e}_j$ is the jth m-dimensional standard unit basis vector. We can also use the mini-batch method here to reduce variance. Assume we have the randomly sampled index set $\mathcal{M}_3 \subseteq [m]$, the mini-batch gradient of $\mathcal{L}$ w.r.t $\boldsymbol{p}$ becomes

$$H(\boldsymbol{w}_t, \boldsymbol{p}_t, f_{\mathcal{M}_3}) = \frac{\beta m}{|\mathcal{M}_3|} \sum_{j \in \mathcal{M}_3} \boldsymbol{e}_j \phi_j(\boldsymbol{w}_t) - \lambda \boldsymbol{p}_t,$$

Then we can perform gradient ascent by using the rule $\boldsymbol{p}_{t+1} = \text{Proj}_{\Delta^m}(\boldsymbol{p}_t + \eta_{\boldsymbol{p}} H(\boldsymbol{w}_t, \boldsymbol{p}_t, f_{\mathcal{M}_3}))$ to update $\boldsymbol{p}$. Note that the projection onto Δ^m can be easily calculated.

The whole algorithm is presented in Algorithm 1.

Algorithm 1 Doubly stochastic zeroth-order gradient method (DSZOG).

```
Input: $T$, $|\mathcal{M}_1|$, $|\mathcal{M}_2|$, $|\mathcal{M}_3|$, $\eta^1_{\boldsymbol{w}}$, $\eta^1_{\boldsymbol{p}}$, $\beta \geq 1$, $q$, $\mu$, $\lambda = 1e - 8$.
Output: $\boldsymbol{w}_T$.
Initialize $\boldsymbol{w}_1$.
Initialize $\boldsymbol{p}_1 = \boldsymbol{p}^*(\boldsymbol{w}_1)$ by solving the strongly concave problem.
for $t = 0, \cdots, T$ do
    Randomly sample $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_q \sim \mathcal{N}(0, \mathbf{1}_d)$.
    Randomly sample a index set $\mathcal{M}_1 \subseteq [n]$ of $\ell_i$.
    Sample a constraint index set $\mathcal{M}_2 \sim \boldsymbol{p} \subseteq [m]$.
    Randomly sample a constraint index set $\mathcal{M}_3 \subseteq [m]$.
    Calculate $G^{\mathcal{L}}_\mu(\boldsymbol{w}_t, \boldsymbol{p}_t, \ell_{\mathcal{M}_1}, f_{\mathcal{M}_2}, \boldsymbol{u}_{[q]}) = \frac{1}{q|\mathcal{M}_1|} \sum_{i \in \mathcal{M}_1} \sum_{k=1}^q G^f_\mu(\boldsymbol{w}_t, \ell_i, \boldsymbol{u}_k) + \frac{1}{q|\mathcal{M}_2|} \sum_{j \in \mathcal{M}_2} \sum_{k=1}^q G^\varphi_\mu(\boldsymbol{w}_t, \boldsymbol{p}_t, f_j, \boldsymbol{u}_k)$.
    Calculate $H(\boldsymbol{w}_t, \boldsymbol{p}_t, f_{\mathcal{M}_3}) = \frac{\beta m}{|\mathcal{M}_3|} \sum_{j \in \mathcal{M}_3} \boldsymbol{e}_j \phi_j(\boldsymbol{w}_t) - \lambda \boldsymbol{p}_t$.
    $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_{\boldsymbol{w}} G^{\mathcal{L}}_\mu(\boldsymbol{w}_t, \boldsymbol{p}_t, \ell_{\mathcal{M}_1}, f_{\mathcal{M}_2}, \boldsymbol{u}_{[q]})$.
    $\boldsymbol{p}_{t+1} = \text{Proj}_{\Delta^m}(\boldsymbol{p}_t + \eta_{\boldsymbol{p}} H(\boldsymbol{w}_t, \boldsymbol{p}_t, f_{\mathcal{M}_3})$.
end for
```

2.4 Accelerated with momentum and variance reduction

To further speed up our method, we modify Algorithm 1 by using exponential moving average (EMA) method Wang et al. (2017); Liu et al. (2020); Cutkosky & Mehta (2020); Guo et al. (2021) and adaptive stepsize. The new algorithm is presented in Algorithm 2.

We first use the following exponential moving average (EMA) method on the zeroth-order and first-order gradient to smooth out short-term fluctuations, highlight longer-term trends and reduce the variance of stochastic gradient Wang et al. (2017); Guo et al. (2021)

$$\boldsymbol{z}^{t+1}_{\boldsymbol{w}} = (1-b)\boldsymbol{z}^t_{\boldsymbol{w}} + b G^{\mathcal{L}}_\mu(\boldsymbol{w}_{t+1}, \boldsymbol{p}_{t+1}, \ell_{\mathcal{M}_1}, f_{\mathcal{M}_2}, \boldsymbol{u}_{[q]}), \ \boldsymbol{z}^{t+1}_{\boldsymbol{p}} = (1-b)\boldsymbol{z}^t_{\boldsymbol{p}} + b H(\boldsymbol{w}_{t+1}, \boldsymbol{p}_{t+1}, f_{\mathcal{M}_3}),$$

where $0 < b < 1$, $\boldsymbol{z}^1_{\boldsymbol{w}} = G^{\mathcal{L}}_\mu(\boldsymbol{w}_1, \boldsymbol{p}_1, \ell_{\mathcal{M}_1}, f_{\mathcal{M}_2}, \boldsymbol{u}_{[q]})$ and $\boldsymbol{z}^1_{\boldsymbol{p}} = H(\boldsymbol{w}_1, \boldsymbol{p}_1, f_{\mathcal{M}_3})$. Here, $H(\boldsymbol{w}_{t+1}, \boldsymbol{p}_{t+1}, f_{\mathcal{M}_3})$ is calculated on the intermediate point $\boldsymbol{p}_{t+1} = (1-a)\boldsymbol{p}_t + a\hat{\boldsymbol{p}}_{t+1}$, which is widely used in Nesterov's momentum method, where $0 < a < 1$ and $\hat{\boldsymbol{p}}_{t+1}$ is the solution of the distribution after updating and projecting onto the Δ^m.

Another modification is the use of adaptive stepsizes of updating $\boldsymbol{w}$ and $\boldsymbol{p}$ which are proportional to $1/\sqrt{\|\boldsymbol{z}^t_{\boldsymbol{w}}\|_2}$ and $1/\sqrt{\|\boldsymbol{z}^t_{\boldsymbol{p}}\|_2}$ Liu et al. (2020); Guo et al. (2021). Therefore, the update rules become

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_{\boldsymbol{w}} \frac{\boldsymbol{z}^t_{\boldsymbol{w}}}{\sqrt{\|\boldsymbol{z}^t_{\boldsymbol{w}}\|_2}} \text{ and } \hat{\boldsymbol{p}}_{t+1} = \text{Proj}_{\Delta^m}(\boldsymbol{p}_t + \eta_{\boldsymbol{p}} \frac{\boldsymbol{z}^t_{\boldsymbol{p}}}{\sqrt{\|\boldsymbol{z}^t_{\boldsymbol{p}}\|_2}}).$$

These two key components of our method, *i.e.,* extrapolation moving average and adaptive stepsize from the root norm of the momentum estimate, make our method enjoy two noticeable benefits: variance reduction of momentum estimate and adaptive convergence.

Algorithm 2 Accelerated doubly stochastic zeroth-order gradient (ADSZOG).

```
Input: $T$, $|\mathcal{M}_1|, |\mathcal{M}_2|, |\mathcal{M}_3|$, $\beta \geq 1$, $q$, $\mu$, $\lambda = 1e-6$, $b \in (0,1)$, $a \in (0,1)$, $\eta_w$ and $\eta_{\boldsymbol{p}}$.
Output: $\boldsymbol{w}_T$.
Initialize $\boldsymbol{w}_1$.
Initialize $\boldsymbol{p}_1 = \boldsymbol{p}^*(\boldsymbol{w}_1)$ by solving the strongly concave problem.
Initialize $\boldsymbol{z}_{\boldsymbol{w}}^1 = G_\mu^{\mathcal{L}}(\boldsymbol{w}_1, \boldsymbol{p}_1, \ell_{\mathcal{M}_1}, f_{\mathcal{M}_2}, \boldsymbol{u}_{[q]})$ and $\boldsymbol{z}_{\boldsymbol{p}}^1 = H(\boldsymbol{w}_1, \boldsymbol{p}_1, f_{\mathcal{M}_3})$.
for $t = 1, \cdots, T$ do
    $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_{\boldsymbol{w}} \frac{\boldsymbol{z}_{\boldsymbol{w}}^t}{\sqrt{\|\boldsymbol{z}_{\boldsymbol{w}}^t\|_2}}$.
    $\hat{\boldsymbol{p}}_{t+1} = \mathrm{Proj}_{\Delta^m}(\boldsymbol{p}_t + \eta_{\boldsymbol{p}} \frac{\boldsymbol{z}_{\boldsymbol{p}}^t}{\sqrt{\|\boldsymbol{z}_{\boldsymbol{p}}^t\|_2}})$.
    $\boldsymbol{p}_{t+1} = (1-a)\boldsymbol{p}_t + a\hat{\boldsymbol{p}}_{t+1}$.
    Randomly sample $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_q \sim \mathcal{N}(0, \mathbf{1}_d)$.
    Randomly sample a index set $\mathcal{M}_1 \subseteq [n]$ of $\ell_i$.
    Sample a constraint index set $\mathcal{M}_2 \sim \boldsymbol{p}_{t+1} \subseteq [m]$.
    Randomly sample a constraint index set $\mathcal{M}_3$.
    Calculate $G_\mu^{\mathcal{L}}(\boldsymbol{w}_{t+1}, \boldsymbol{p}_{t+1}, \ell_{\mathcal{M}_1}, f_{\mathcal{M}_2}, \boldsymbol{u}_{[q]}) = \frac{1}{q|\mathcal{M}_1|}\sum_{i\in\mathcal{M}_1}\sum_{k=1}^q G_\mu^f(\boldsymbol{w}_{t+1}, \ell_i, \boldsymbol{u}_k) + \frac{1}{q|\mathcal{M}_2|}\sum_{j\in\mathcal{M}_2}\sum_{k=1}^q G_\mu^\varphi(\boldsymbol{w}_{t+1}, \boldsymbol{p}_{t+1}, f_j, \boldsymbol{u}_k)$.
    Calculate $H(\boldsymbol{w}_{t+1}, \boldsymbol{p}_{t+1}, f_{\mathcal{M}_3}) = \frac{\beta m}{|\mathcal{M}_3|}\sum_{j\in\mathcal{M}_3} \boldsymbol{e}_j \phi_j(\boldsymbol{w}_{t+1}) - \lambda \boldsymbol{p}_{t+1}$.
    $\boldsymbol{z}_{\boldsymbol{w}}^{t+1} = (1-b)\boldsymbol{z}_{\boldsymbol{w}}^t + bG_\mu^{\mathcal{L}}(\boldsymbol{w}_{t+1}, \boldsymbol{p}_{t+1}, \ell_{\mathcal{M}_1}, f_{\mathcal{M}_2}, \boldsymbol{u}_{[q]})$.
    $\boldsymbol{z}_{\boldsymbol{p}}^{t+1} = (1-b)\boldsymbol{z}_{\boldsymbol{p}}^t + bH(\boldsymbol{w}_{t+1}, \boldsymbol{p}_{t+1}, f_{\mathcal{M}_3})$.
end for
```

3 Convergence Analysis

In this section, we discuss the convergence performance of our methods. The detailed proofs are given in the appendix.

3.1 Stationary point

In this subsection, we first give the assumption about $\mathcal{L}$ which is also used in Wang et al. (2020); Huang et al. (2020a) and then give the definitions of the stationary point.

Assumption 1 *The objective function $\mathcal{L}(\boldsymbol{w}, \boldsymbol{p})$ has the following properties:*

1. *$\mathcal{L}(\boldsymbol{w}, \boldsymbol{p})$ is continuously differentiable in $\boldsymbol{w}$ and $\boldsymbol{p}$. $\mathcal{L}(\boldsymbol{w}, \boldsymbol{p})$ is nonconvex with respect to $\boldsymbol{w}$, and $\mathcal{L}(\boldsymbol{w}, \boldsymbol{p})$ is τ-strongly concave with respect to $\boldsymbol{p}$.*
2. *The function $g(\boldsymbol{w}) := \max_{\boldsymbol{p}} \mathcal{L}(\boldsymbol{w}, \boldsymbol{p})$ is lower bounded, and $g(\boldsymbol{w})$ is L_g-Lipschitz continuous.*
3. *When viewed as a function in $\mathbb{R}^{d+m}$, $\mathcal{L}(\boldsymbol{w}, \boldsymbol{p})$ is L-gradient Lipschitz. That is there exists constant $L > 0$ such that $\|\nabla\mathcal{L}(\boldsymbol{w}_1, \boldsymbol{p}_1) - \nabla\mathcal{L}(\boldsymbol{w}_2, \boldsymbol{p}_2)\|_2 \leq L\|(\boldsymbol{w}_1, \boldsymbol{p}_1) - (\boldsymbol{w}_2, \boldsymbol{p}_2)\|_2$ and let $\kappa := L/\tau$ and $\kappa > 1$.*

For a general nonconvex constrained optimization problem, the stationary point Lin et al. (2019) is defined as follows,

Definition 1 *$\boldsymbol{w}^*$ is said to be the stationary point of problem (1), if the following conditions holds,*

$$\nabla_{\boldsymbol{w}} f_0(\boldsymbol{w}^*) + \sum_{j=1}^m \alpha_j^* \nabla_{\boldsymbol{w}} f_j(\boldsymbol{w}^*) = \mathbf{0}, \quad f_j(\boldsymbol{w}^*) \leq 0, \quad \alpha_j^* f_j(\boldsymbol{w}^*) = 0, \quad \forall i \in \{1, \cdots, m\},$$

where $\boldsymbol{\alpha}^* = [\alpha_1, \cdots, \alpha_m]_t$ denotes the Lagrangian multiplier and $\alpha_j \geq 0, \forall i = 1, \cdots, m$.

However, it is hard to compute a solution that satisfies the above conditions exactly Lin et al. (2019). Therefore, finding the following ϵ-stationary point Lin et al. (2019) is more practicable,

Definition 2 *(ϵ-stationary) $\boldsymbol{w}^*$ is said to be the ϵ-stationary point of problem (1), if there exists a vector $\boldsymbol{\alpha}^* \geq \mathbf{0}$, such that the following conditions hold,*

$$\|\nabla_{\boldsymbol{w}} f_0(\boldsymbol{w}^*) + \sum_{j=1}^{m} \alpha_j^* \nabla_{\boldsymbol{w}} f_j(\boldsymbol{w}^*)\|_2^2 \leq \epsilon_1^2,\ \sum_{j=1}^{m} (\max\{f_j(\boldsymbol{w}^*), 0\})^2 \leq \epsilon_2^2,\ \sum_{j=1}^{m} (\alpha_j f_j(\boldsymbol{w}^*))^2 \leq \epsilon_3^2.$$

Since we reformulate the constrained problem as a minimax problem, here we give the definition of the approximation stationary point of the minimax problem and then show its relationship with Definition 2. According to Wang et al. (2020), we have the following definition and proposition,

Definition 3 *A point $(\boldsymbol{w}^*, \boldsymbol{p}^*)$ is called the ϵ-stationary point of problem $\min_{\boldsymbol{w}} \max_{\boldsymbol{p} \in \Delta^m} \mathcal{L}(\boldsymbol{w}, \boldsymbol{p})$ if it satisfies the conditions: $\|\nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \boldsymbol{p})\|_2^2 \leq \epsilon^2$ and $\|\nabla_{\boldsymbol{p}} \mathcal{L}(\boldsymbol{w}, \boldsymbol{p})\|_2^2 \leq \epsilon^2$.*

Proposition 1 *If Assumption 1 holds, $\dfrac{2\epsilon^2 + 2m\lambda^2}{\beta^2} \leq \epsilon_2^2$ and $(\boldsymbol{w}^*, \boldsymbol{p}^*)$ is the ϵ-stationary point defined in Definition 3 of the problem $\min_{\boldsymbol{w}} \max_{\boldsymbol{p} \in \Delta^m} \mathcal{L}(\boldsymbol{w}, \boldsymbol{\alpha})$, then $\boldsymbol{w}^*$ is the ϵ-stationary point defined in Definition 2 of the constrained problem 1.*

As proposed in Wang et al. (2020), the minimax problem 2 is equivalent to the following minimization problem:

$$\min_{\boldsymbol{w}} \left\{ g(\boldsymbol{w}) := \max_{\boldsymbol{p} \in \Delta^m} \mathcal{L}(\boldsymbol{w}, \boldsymbol{p}) = \mathcal{L}(\boldsymbol{w}, \boldsymbol{p}^*(\boldsymbol{w})) \right\} \tag{3}$$

where $\boldsymbol{p}^*(\boldsymbol{w}) = \arg\max_{\boldsymbol{p}} \mathcal{L}(\boldsymbol{w}, \boldsymbol{p})$. Here, we give stationary point the minimization problem 3 and its relationship with Definition 3 as follows,

Definition 4 *We call $\boldsymbol{w}^*$ an ϵ-stationary point of a differentiable function $g(\boldsymbol{w})$, if $\|\nabla g(\boldsymbol{w}^*)\|_2 \leq \epsilon$.*

Proposition 2 *Under Assumption 1, if a point $\mathbf{w}'$ is an ϵ-stationary point in terms of Definition 4, then an ϵ-stationary point $\boldsymbol{w}^*, \boldsymbol{p}^*$ in terms of Definition 3 can be obtained.*

Remark 1 *According to Proposition 1 and Proposition 2, we have that once we find the ϵ-stationary point in terms of Definition 4, then we can get the ϵ-stationary point in terms of Definition 2.*

3.2 Convergence Rate of the DSZOG

Here, we present some assumptions used in our analysis, which are widely used in the convergence analysis.

Assumption 2 *For any $\boldsymbol{w} \in \mathbb{R}^d$, the following properties holds, $\mathbb{E}[G_\mu^{\mathcal{L}}(\boldsymbol{w}, \boldsymbol{p}, \ell_{\mathcal{M}_1}, f_{\mathcal{M}_2}, \boldsymbol{u}_{[q]})] = \nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \boldsymbol{p})$, $\mathbb{E}[H(\boldsymbol{w}, \boldsymbol{p}, f_{\mathcal{M}_3})] = \nabla_{\boldsymbol{p}} \mathcal{L}(\boldsymbol{w}, \boldsymbol{p})$, $\mathbb{E}[\|G_\mu^{\mathcal{L}}(\boldsymbol{w}_1, \boldsymbol{p}_1, \ell_{\mathcal{M}_1}, f_{\mathcal{M}_2}, \boldsymbol{u}_{[q]}) - \nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}_1, \boldsymbol{p}_1)\|_2] \leq \sigma_1^2$ and $\mathbb{E}[\|H(\boldsymbol{w}, \boldsymbol{p}, f_{\mathcal{M}_3}) - \nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}_t, \boldsymbol{p}_t)\|_2] \leq \sigma_2^2$.*

Let $\mathcal{L}_\mu(\boldsymbol{w}, \boldsymbol{p}) = \mathbb{E}_{\boldsymbol{u}}[\mathcal{L}(\boldsymbol{w} + \mu \boldsymbol{u}, \boldsymbol{p})]$ and $\boldsymbol{u} \sim \mathcal{N}(0, \mathbf{1}_d)$. Following the theoretical analysis in Wang et al. (2018), we have the following theorem,

Theorem 1 *Under Assumptions 1 and 2, by setting $\eta_{\boldsymbol{w}} = \dfrac{1}{4 \times 16^2 \kappa^2 (\kappa+1)^2 (L+1)}$, $\eta_{\boldsymbol{p}} = \dfrac{1}{2L}$, $\mu := \mathcal{O}(\epsilon d^{-3/2} L^{-2})$, $T > \max\{\dfrac{2(g(\boldsymbol{w}_0) - g(\boldsymbol{w}_{T+1}))}{0.9325 \epsilon^2 \eta_{\boldsymbol{w}}}, \dfrac{\sigma_1^2}{16 \kappa \eta_w^2 \epsilon^2}\}$ and $\boldsymbol{p}_0 = \boldsymbol{p}^*(\boldsymbol{w}_0)$, our algorithm DSZOG has $g(\boldsymbol{w}) = \max_{\boldsymbol{p} \in \mathcal{P}} \mathcal{L}(\boldsymbol{w}, \boldsymbol{p})$, i.e., $\dfrac{1}{T+1} \sum_{t=0}^{T} \mathbb{E}[\|\nabla_{\boldsymbol{w}} g(\boldsymbol{w}_t)\|_2^2] \leq \epsilon^2$.*

Remark 2 *Based on Proposition 1 and 2, Theorem 1 shows that our method can finally converge to the ϵ-stationary point of the constrained problem 1 at the rate of $\mathcal{O}(L^5/T)$ by setting the learning rate $\eta_{\boldsymbol{w}} = \dfrac{1}{4 \times 16^2 \kappa^2 (\kappa+1)^2 (L+1)}$ and $\kappa = L/\tau$.*

3.3 Convergence Rate of the Accelerated Method

Similar to Guo et al. (2021), we assume $1/\sqrt{\|\boldsymbol{z}_{\boldsymbol{p}}^t\|_2}$ and $1/\sqrt{\|\boldsymbol{z}_{\boldsymbol{w}}^t\|_2}$ are bounded as follows,

Assumption 3 *We have* $c_{1,l} \leq \dfrac{1}{\sqrt{\|\boldsymbol{z}_{\boldsymbol{w}}^t\|_2}} \leq c_{1,u}$ *and* $c_{2,l} \leq \dfrac{1}{\sqrt{\|\boldsymbol{z}_{\boldsymbol{p}}^t\|_2}} \leq c_{2,u}$

Then, following the framework in Guo et al. (2021); Wang et al. (2018); Huang et al. (2020a), we have the following theorem,

Theorem 2 *Under Assumptions 1, 2 and 3, if* $a \leq 1$, $\tau \leq L$,$\boldsymbol{p}^*(w_1) = \boldsymbol{p}_1$, $\boldsymbol{z}_{\boldsymbol{p}}^1 = H(\boldsymbol{w}_t, \boldsymbol{p}_t, f_{\mathcal{M}_3})$, $\boldsymbol{z}_{\boldsymbol{w}}^1 = G_\mu^{\mathcal{L}}(\boldsymbol{w}_t, \boldsymbol{p}_t, \ell_{\mathcal{M}_1}, f_{\mathcal{M}_2}, \boldsymbol{u}_{[q]})$, $\eta_p \leq \min\{\frac{1}{3c_{2.l}L}, \frac{b^2}{\tau a^2 c_{2,l}}, \frac{\tau b^2}{32L^2a^2c_{2,l}}\}$, $\eta_w^2 \leq \min\{\frac{c_{1,l}^2}{4Lc_{1,u}^4}, \frac{b^2}{4c_{1,u}^2L^2}, \frac{\tau^2a^2\eta_p^2c_{2,l}^2}{128L_g^2L^2c_{1,u}}, \frac{\tau^2b^2}{128L^4c_{1,u}^2}\}$, $\mu \leq \frac{\epsilon}{L(d+3)^{3/2}}$, $b \leq \min\{\frac{\epsilon^2}{2\sigma_1^2}, \frac{\tau^2\epsilon^2}{64\sigma_2^2L^2}\}$ *and* $T \geq \max\{\frac{2(g(\boldsymbol{w}_1) - g(\boldsymbol{w}_T))}{\epsilon^2\eta_w c_{1,l}}, \frac{2\sigma_1^2}{\epsilon^2 b}, \frac{64\sigma_2^2L^2}{\epsilon^2\tau^2 b}\}$, *we have* $\frac{1}{T}\mathbb{E}[\sum_{t=1}^T \|\nabla g(\boldsymbol{w}_t)\|_2^2] \leq \epsilon^2$.

Remark 3 *Based on Proposition 1 and 2, Theorem 2 shows that our method can finally converge to the* ϵ*-stationary point of the constrained problem 1. More importantly, by choosing* $\eta_{\boldsymbol{w}} \propto \mathcal{O}(1/L^6)$ *and* $b \propto \mathcal{O}(1/L^6)$*, our proposed ADSZOG has the convergence rate of* $\mathcal{O}(L^6/T)$*. Obviously, the convergence rate of ADSZOG is faster than DSZOG.*

4 Experiments

4.1 Baselines

In this subsection, we summarized the baselines used in our experiments as follows,

1. **ZOPSGD**Liu et al. (2018b). In each iteration, ZOPSGD calculates the stochastic zeroth-order gradient of f_0 to update the parameters and then solve a constrained quadratic problem to project the solution into the feasible set.
2. **ZOSCGD**Balasubramanian & Ghadimi (2018). In each iteration, ZOSCGD calculates the stochastic zeroth-order gradient of f_0 and then use the conditional gradient method to update the parameters by solving a constrained linear problem.

Note that both ZOPSGD and ZOSCGD are designed for solving the constrained problem with white-box constraints. However our methods can solve the problem with nonconvex/convex white/black-box constraints. To compare the performance of our methods with ZOPSGD and ZOSCGD, we design two problem with white-box constraints in the next subsection.

4.2 Applications

In this subsection, we give the introduction of the applications used in our experiments.

Classification with Pairwise Constraints We evaluate the performance of all the methods on the binary classification with pairwise constraints learning problem. Given a set of training samples $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \{+1, -1\}$. In this task, we learn a linear model $h(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{x}^T\boldsymbol{w}$ to classify the dataset and ensure the any positive sample $\boldsymbol{x}_i^+ \in \mathcal{D}^+ := \{(\boldsymbol{x}_i, +1)\}_{i=1}^{n_p}$ has larger function value than the negative sample $\boldsymbol{x}_j^- \in \mathcal{D}^- := \{(\boldsymbol{x}_j, +1)\}_{i=1}^{n_n}$, where n_p and n_n denotes the number of positive samples and negative samples, respectively. Then, we can formulate this problem as follows,

$$\min_{\boldsymbol{w}} \frac{1}{n}\sum_{i=1}^n \ell(h(\boldsymbol{x}_i, \boldsymbol{w}), y_i),\ s.t.\ h(\boldsymbol{x}_i^+, \boldsymbol{w}) - h(\boldsymbol{x}_j^-, \boldsymbol{w}) \geq 0,\ \forall \boldsymbol{x}_i^+ \in \mathcal{D}^+\ \boldsymbol{x}_j^- \in \mathcal{D}^-$$

where $\ell(u, v) = c^2(1 - \exp(-\dfrac{(v-u)^2}{c^2}))$ is viewed as a black-box function. We summarized the datasets used in this application in Table 2. We randomly sample 1000 data samples from

Table 2: Datasets used in classification with pairwise constraints (We give the approximate size of constraints).

Data	Dimension	Constriants
w8a	300	≃8000
a9a	123	≃40000
gen	50	≃60000
svmguide3	22	≃40000

Table 3: Test accuracy (%) of all the methods in classification with pairwise constraints.

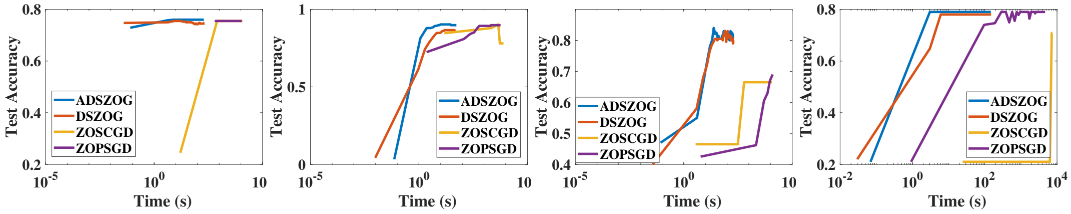
Data	ADSZOG	DSZOG	ZOSCGD	ZOPSGD
a9a	**75.90** ± 0.26	75.37 ± 0.55	75.35 ± 0.13	75.37 ± 0.19
w8a	**89.94** ± 0.28	86.62 ± 0.93	83.53 ± 0.58	89.02 ± 0.97
gen	**82.33** ± 0.76	82.11 ± 0.28	66.33 ± 0.07	66.83 ± 0.57
svmguide3	**79.56** ± 0.49	78.83 ± 0.90	71.21 ± 0.57	78.63 ± 0.26

the original datasets, and then divide all the datasets into 3 parts, i.e., 50% for training, 30% for testing and 20% for validation. We fix the batch size of data sample at 128 for all the methods and $|\mathcal{M}_2| = |\mathcal{M}_3| = 128$. The learning rates of all the methods are chosen from $\{0.01, 0.001, 0.0001\}$. In our methods, the penalty parameter β is chosen from $\{0.1, 1, 10\}$, a and b are chosen from $\{0.1, 0.5, 0.9\}$ on the validation sets.

Classification with Fairness Constraints. In this problem, we consider the binary classification problem with a large amount of fairness constraints Zafar et al. (2017). Given a set of training samples $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$. In this task, we learn a linear model $h(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{x}^T \boldsymbol{w}$. Assume that each sample has an associate sensitive feature vector $\boldsymbol{z} \in \mathbb{R}^r$. We denote $z_{ij} \in \{0, 1\}$ as the j-th sensitive feature of i-th sample. The classifier h cannot use the protected characteristic $\boldsymbol{z}$ at decision time, as it will constitute an unfair treatment. A number of metrics have been used to determine how fair a classifier is with respect to the sensitive features. According to Zafar et al. (2017), the fair classification problems can be formulated as follows,

$$\min_{\boldsymbol{w}} \frac{1}{n} \sum_{i=1}^{n} \ell(h(\boldsymbol{x}_i, \boldsymbol{w}), y_i) \ s.t. \frac{1}{n} \sum_{i=1}^{n} (z_{ij} - \bar{z}_j) g(y_i, \boldsymbol{x}_i) \leq c, \ \frac{1}{n} \sum_{i=1}^{n} (z_{ij} - \bar{z}_j) g(y_i, \boldsymbol{x}_i) \geq -c,$$

where $j = 1, \cdots, r$, $\ell(u, v)$ denotes the loss functions, c is the covariance threshold which specifies an upper bound on the covariance between the sensitive attributes $\boldsymbol{z}$ and the signed distance $g(y, \boldsymbol{x})$. We use the hinge loss $\ell(u, v) = \max\{1 - uv, 0\}$ in this experiment and we view it as a black-box function. In addition, we use the following two functions to build the fairness constraints, $g(y, \boldsymbol{x}) = \left\{ \begin{array}{c} \min\{0, \frac{1+y}{2} y h(\boldsymbol{x}, \boldsymbol{w})\} \\ \min\{0, \frac{1-y}{2} h(\boldsymbol{x}, \boldsymbol{w})\} \end{array} \right.$. Since the datasets with multiple sensitive features are difficult to find, we generate 4 datasets with 2000 samples in this task and summarize them in Table 4. For each dataset, we randomly choose several features to be the sensitive features, and then separate



(a) a9a (b) w8a (c) gen (d) svmguide3

Figure 1: Test accuracy against training time of all the methods in classification with pairwise constraints (We stop the algorithms if the training time is more than 10000 seconds).

Table 4: Datasets used in classification with fairness constraints.

Data	Dimension	Sensitive Features	Constraints
D1	100	10	40
D2	200	20	80
D3	300	20	80
D4	400	20	80

Table 5: Test accuracy (%) of all the methods in classification with fairness constraints.

Data	ADSZOG	DSZOG	ZOSCGD	ZOPSGD
D1	**87.33** $\pm$ 0.38	86.83 $\pm$ 0.52	51.08 $\pm$ 0.57	59.16 $\pm$ 0.37
D2	**84.75** $\pm$ 0.25	84.02 $\pm$ 0.31	69.70 $\pm$ 0.24	68.00 $\pm$ 0.54
D3	**83.58** $\pm$ 0.14	82.00 $\pm$ 0.05	66.33 $\pm$ 0.30	66.84 $\pm$ 0.57
D4	**64.91** $\pm$ 0.94	64.50 $\pm$ 0.25	52.16 $\pm$ 0.38	55.25 $\pm$ 0.90

them into 3 parts, i.e., 50% for training, 30% for testing and 20% for validation. We fix the batch size of data sample at 128 for all the methods and $|\mathcal{M}_2| = |\mathcal{M}_3| = 10$. The learning rates of all the methods are chosen from $\{0.01, 0.001, 0.0001\}$. For our methods, the penalty parameter β is chosen from $\{0.1, 1, 10\}$, a and b are chosen from $\{0.1, 0.5, 0.9\}$ on the validation sets.

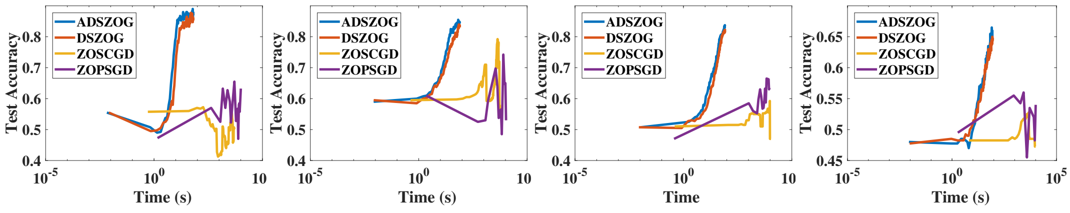
We run all the methods 10 times on a 3990x workstation.

4.3 Results and Discussion

We present the results in Figures 1, 2 and Tables 3, 5. Note that for ZOSCGD and ZOPSGD, if the training time is larger than 10000 seconds, the algorithms are stopped. From Tables 3 and 5, we can find that our methods ADSZOG and DSZOG have the highest test accuracy in most cases in both two applications. In addition, from Figures 1 and 2, we can find that our methods are faster than ZOSCGD and ZOPSGD. This is because ZOSCGD and ZOPSGD need to solve a subproblem with a large number of constraints in each iteration and the existing Python package cannot efficiently deal with such a problem. What's worse, ZOPSGD and ZOSCGD focus on solving the problem with convex constraints while the constraints in the fairness problem are nonconvex. This makes ZOPSGD and ZOSCGD cannot find the stationary point. However, by using the penalty framework, our methods can still converge to the stationary point when the constraints are nonconvex. In addition, by using a stochastic manner on the constraint, our method can efficiently deal with a large number of convex/nonconvex constraints. In addition, we can also find that ADSZOG can converge faster than DSZOG. This is because we use the exponential moving average in ADSZOG which makes it benefits from variance reduction and momentum acceleration. All these results demonstrate that our method is superior to ZOSCGD and ZOPSGD in the heavily constrained nonconvex problem.

5 Conclusion

In this paper, we propose two efficient ZO method to solve the heavily constrained nonconvex black-box problem, i.e., DSZOG and ADSZOG. We also give the convergence analysis of our proposed methods. The experimental results on two applications demonstrate the superiority of our method in terms of accuracy and training time .



(a) D1 (b) D2 (c) D3 (d) D4

Figure 2: Test accuracy against training time of all the methods in classification with fairness constraints (We stop the algorithms if the training time is more than 10000 seconds).

References

Alekh Agarwal, Ofer Dekel, and Lin Xiao. Optimal algorithms for online convex optimization with multi-point bandit feedback. In *COLT*, pp. 28–40. Citeseer, 2010.

Krishnakumar Balasubramanian and Saeed Ghadimi. Zeroth-order (non)-convex stochastic optimization via conditional gradient and gradient updates. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 3459–3468, 2018.

Krishnakumar Balasubramanian and Saeed Ghadimi. Zeroth-order nonconvex stochastic optimization: Handling constraints, high dimensionality, and saddle points. *Foundations of Computational Mathematics*, pp. 1–42, 2021.

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pp. 15–26, 2017.

Krzysztof Choromanski, Mark Rowland, Vikas Sindhwani, Richard Turner, and Adrian Weller. Structured evolution with compact architectures for scalable policy optimization. In *International Conference on Machine Learning*, pp. 970–978. PMLR, 2018.

Kenneth L Clarkson, Elad Hazan, and David P Woodruff. Sublinear optimization for machine learning. *Journal of the ACM (JACM)*, 59(5):1–49, 2012.

Andrew Cotter, Maya Gupta, and Jan Pfeifer. A light touch for heavily constrained sgd. In *Conference on Learning Theory*, pp. 729–771. PMLR, 2016.

Ashok Cutkosky and Harsh Mehta. Momentum improves normalized sgd. In *International Conference on Machine Learning*, pp. 2260–2268. PMLR, 2020.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: near-optimal non-convex optimization via stochastic path integrated differential estimator. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 687–697, 2018.

Hongchang Gao and Heng Huang. Can stochastic zeroth-order frank-wolfe method converge faster for non-convex problems? In *International Conference on Machine Learning*, pp. 3377–3386. PMLR, 2020.

Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Bin Gu, Zhouyuan Huo, and Heng Huang. Zeroth-order asynchronous doubly stochastic algorithm with variance reduction. *arXiv preprint arXiv:1612.01425*, 2016.

Zhishuai Guo, Yi Xu, Wotao Yin, Rong Jin, and Tianbao Yang. On stochastic moving-average estimators for non-convex optimization. *arXiv preprint arXiv:2104.14840*, 2021.

Feihu Huang, Shangqian Gao, Jian Pei, and Heng Huang. Accelerated zeroth-order and first-order momentum methods from mini to minimax optimization. *arXiv preprint arXiv:2008.08170*, 2020a.

Feihu Huang, Lue Tao, and Songcan Chen. Accelerated stochastic gradient-free and projection-free methods. In *International Conference on Machine Learning*, pp. 4519–4530. PMLR, 2020b.

Qihang Lin, Runchao Ma, and Yangyang Xu. Inexact proximal-point penalty methods for constrained non-convex optimization. *arXiv preprint arXiv:1908.11518*, 2019.

Tianyi Lin, Chi Jin, and Michael Jordan. On gradient descent ascent for nonconvex-concave minimax problems. In *International Conference on Machine Learning*, pp. 6083–6093. PMLR, 2020.

Mingrui Liu, Wei Zhang, Francesco Orabona, and Tianbao Yang. Adam $^+$: A stochastic method with adaptive variance reduction. *arXiv preprint arXiv:2011.11985*, 2020.

Sijia Liu, Bhavya Kailkhura, Pin-Yu Chen, Paishun Ting, Shiyu Chang, and Lisa Amini. Zeroth-order stochastic variance reduction for nonconvex optimization. *Advances in Neural Information Processing Systems*, 31:3727–3737, 2018a.

Sijia Liu, Xingguo Li, Pin-Yu Chen, Jarvis Haupt, and Lisa Amini. Zeroth-order stochastic projected gradient descent for nonconvex optimization. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 1179–1183. IEEE, 2018b.

Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *International Conference on Machine Learning*, pp. 2613–2621. PMLR, 2017.

Mengdi Wang, Ethan X Fang, and Han Liu. Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions. *Mathematical Programming*, 161(1-2): 419–449, 2017.

Yining Wang, Simon Du, Sivaraman Balakrishnan, and Aarti Singh. Stochastic zeroth-order optimization in high dimensions. In *International Conference on Artificial Intelligence and Statistics*, pp. 1356–1365. PMLR, 2018.

Zhongruo Wang, Krishnakumar Balasubramanian, Shiqian Ma, and Meisam Razaviyayn. Zeroth-order algorithms for nonconvex minimax problems with improved complexities. *stat*, 1050:22, 2020.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th international conference on world wide web*, pp. 1171–1180, 2017.