Minibatch Stochastic Three Points Method for Unconstrained Smooth Minimization

Soumia Boucherouite[1], **Grigory Malinovsky**[2], **Peter Richtárik**[2], **EL Houcine Bergou**[1]

[1]College of Computing, Mohammed VI Polytechnic University, Ben Guerir, Morocco
[2]King Abdullah University of Science and Technology, Thuwal, Saudi Arabia
{soumia.boucherouite,elhoucine.bergou}@um6p.ma, {grigorii.malinovskii,peter.richtarik}@kaust.edu.sa

Abstract

We present a new zero-order optimization method called *Minibatch Stochastic Three Points* (`MiSTP`), specifically designed to solve stochastic unconstrained minimization problems when only an approximate evaluation of the objective function is possible. `MiSTP` is an extension of the Stochastic Three Point Method (`STP`) proposed by Bergou, Gorbunov, and Richtárik (2020). The key innovation of `MiSTP` is that it selects the next point solely based on the objective function approximation, without relying on its exact evaluation. At each iteration, `MiSTP` generates a random search direction and compares the approximations of the objective function at the current point, the randomly generated direction and its opposite. The best of these three points is chosen as the next iterate. We analyze the worst-case complexity of `MiSTP` in the convex and non-convex cases and demonstrate that it matches the most accurate complexity bounds known in the literature for zero-order optimization methods. We perform extensive numerical evaluations to assess the computational efficiency of `MiSTP` and compare its performance to other state-of-the-art methods by testing it on several machine learning tasks. The results show that `MiSTP` outperforms or has comparable performance against state-of-the-art methods indicating its potential for a wide range of practical applications.

1 Introduction

In this paper, we consider the following unconstrained finite-sum optimization problem:

$$\min_{x\in\mathbb{R}^d} f(x) \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} f_i(x) \tag{1}$$

where each $f_i : \mathbb{R}^d \to \mathbb{R}$ is a smooth objective function. Such kind of problems arises in a large body of machine learning (ML) applications including logistic regression (Conroy and Sajda 2012), ridge regression (Shen et al. 2013), least squares problems (Suykens and Vandewalle 1999), and deep neural networks training. The formulation (1) can express the distributed optimization problem across n agents, where each function f_i represents the objective function of agent i, or the optimization problem where each f_i is the objective function associated with the data point i.

We assume that we work in the Zero Order (ZO) optimization settings, i.e., we do not have access to the derivatives of any function f_i and only functions evaluations are available. Such situation arises in many fields and may occur due to multiple reasons, for example: (i) In many optimization problems, there is only availability of the objective function as the output of a black-box or simulation oracle and hence the absence of derivative information (Conn, Scheinberg, and Vicente 2009). (ii) There are situations where the objective function evaluation is done through an old software. Modification of this software to provide first-order derivatives may be too costly or impossible (Conn, Scheinberg, and Vicente 2009; Nesterov and Spokoiny 2017). (iii) In some situations, derivatives of the objective function are not available but can be extracted. This necessitates access and a good understanding of the simulation code. This process is considered invasive to the simulation code and also very costly in terms of coding efforts (Kramer, Ciaurri, and Koziel 2011). (iv) In the case of using a commercial software that evaluates only the functions, it is impossible to compute the derivatives because the simulation code is inaccessible (Kramer, Ciaurri, and Koziel 2011; Conn, Scheinberg, and Vicente 2009). (v) In the case of having access only to noisy function evaluations, computing derivatives is useless because they are unreliable (Conn, Scheinberg, and Vicente 2009). ZO optimization has been used in many ML applications, for instance: hyperparameters tuning of ML models (Turner et al. 2021; P.Koch et al. 2018), multi-agent target tracking (Al-Abri et al. 2021), policy optimization in reinforcement learning algorithms (Malik et al. 2020; Li et al. 2022), maximization of the area under the curve (AUC) (Ghanbari and Scheinberg 2017), automatic speech recognition (Watanabe and Roux 2014), and the generation of black-box adversarial attacks on deep neural network classifiers (Ughi, Abrol, and Tanner 2021). Google Vizier system (Golovin et al. 2017) which is the de facto parameter tuning engine at Google is also based on ZO optimization.

One way to solve problem (1) is to approximate the gradient using a gradient estimation technique based only on function values, and then apply a first-order optimization method. Nesterov and Spokoiny (2017) extended the Gradient Descent (`GD`) algorithm to the ZO setting and proposed `RGF` (also called `ZO-GD`) in which the full gradient is replaced by a two-point random gradient estimation. Ghadimi and Lan (2013) further extended RGF to the stochastic setting and proposed RSGF (also called ZO-SGD). To reduce the variance of the gradient estimates and further improve the convergence rate of ZO-SGD, Liu et al. (2018) proposed ZO-SVRG which is based on the minibatch variant of SVRG method (Reddi et al. 2016) where they replaced the gradient by its estimation computed using the two-point gradient estimator. The authors further proposed two accelerated versions of ZO-SVRG which are ZO-SVRG-Ave and ZO-SVRG-Coord that uses the average random gradient estimator and coordinate-wise gradient estimator respectively.

Another popular class of ZO methods is Direct-Search (DS) methods. They determine the next iterate based solely on function values and does not develop an approximation of the derivatives or build a surrogate model of the the objective function (Conn, Scheinberg, and Vicente 2009). For a comprehensive view about classes of ZO methods we refer the reader to a survey by Larson, Menickelly, and Wild (2019). More related to our work, Bergou, Gorbunov, and Richtárik (2020) proposed a ZO method called Stochastic Three Points (STP) which is a general variant of direct search methods. At each training iteration, STP generates a random search direction s according to a certain probability distribution and updates the iterate as follow:

$$x = \arg\min\{f(x - \alpha s), f(x + \alpha s), f(x)\}$$

where $\alpha > 0$ is the stepsize. STP is simple, very easy to implement, and has better complexity bounds than deterministic direct search (DDS) methods. Due to its efficiency and simplicity, STP paved the way for other interesting works that are conducted for the first time, namely the first work on importance sampling in the random direct search setting ($\mathtt{STP}_{IS}$ method) (Bibi et al. 2020) and the first ZO method with heavy ball momentum (SMTP) and with importance sampling ($\mathtt{SMTP}_{IS}$) (Gorbunov et al. 2020). To solve problem (1), STP evaluates f two times at each iteration, which means performing two new computations using all the training data for one update of the parameters. In fact, proceeding in such manner is not all the time efficient. In cases when the total number of training samples is extremely large, such as in the case of large scale machine learning, it becomes computationally expensive to use all the dataset at each iteration of the algorithm. Moreover, training an algorithm using minibatches of the data could be as efficient or better than using the full batch as in the case of SGD (Gower et al. 2019). Motivated by this, we introduced MiSTP to extend STP to the case of using subsets of the data at each iteration of the training process.

We consider in this paper the finite-sum problem as it is largely encountered in ML applications, but our approach is applicable to the more general case where we do not have necessarily the finite-sum structure and only an approximation of the objective function can be computed. Such situation may happen, for instance, in the case where the objective function is the output of a stochastic oracle that provides only noisy/stochastic evaluations.

Contributions

In this section, we highlight the key contributions of this work.

- We propose MiSTP method to extend the STP method (Bergou, Gorbunov, and Richtárik 2020) to the case of using only an approximation of the objective function at each iteration.
- We analyse our method's complexity in the case of nonconvex and convex objective function.
- We present experimental results of the performance of MiSTP on multiple ML tasks, namely on ridge regression, regularized logistic regression, training of a neural network, and the generation of a black box adversarial attack on a deep neural network classifier. We evaluate the performance of MiSTP with different minibatch sizes and in comparison with other ZO methods.

Outline

The paper is organized as follow: In Section 2, we present our MiSTP method. In Section 2.1, we describe the main assumptions on the random search directions which ensure the convergence of our method. These assumptions are the same as the ones used for STP (Bergou, Gorbunov, and Richtárik 2020). Then, in Section 2.2, we formulate the key lemma for the iteration complexity analysis. In Section 3, we analyze the worst case complexity of our method for smooth nonconvex and convex problems. In Section 4, we present and discuss our experiments results. In Section 4.1, we report the results on ridge regression and regularized logistic regression problems, in Section 4.2, we report the result on neural networks, and in Section 4.3, we present the results on the task of generating black-box adversarial attacks on a deep neural network classifier. Finally, we conclude in Section 5.

Notation

Throughout the paper, $\mathcal{D}$ will denote a probability distribution over $\mathbb{R}^d$. We use $\mathbf{E}\left[\cdot\right]$ to denote the expectation, $\mathbf{E}_{\xi}\left[\cdot\right]$ to denote the expectation over the randomness of ξ conditional to other random quantities, and for two random variables X and Y, $\mathbf{E}[X|Y]$ denotes the expectation of X given Y. $\langle x, y\rangle = x^\top y$ corresponds to the inner product of x and y. We denote also by $\|\cdot\|_2$ the ℓ_2-norm, and by $\|\cdot\|_{\mathcal{D}}$ a norm dependent on $\mathcal{D}$. We denote by $f_{\mathcal{B}}$:

$$f_{\mathcal{B}}(x) = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} f_i(x), \tag{2}$$

where $\mathcal{B}$ is a subset on indexes chosen from the set $[1, 2, \ldots, n]$ and $|\mathcal{B}|$ is its cardinal.

2 MiSTP method

Our *minibatch stochastic three points* (MiSTP) algorithm is formalized below as Algorithm 1.

Due to the randomness of the search directions s_k and the minibatches $\mathcal{B}_k$ for $k \geq 0$, the iterates are also random vectors for all $k \geq 1$. The starting point x_0 is not random (the initial objective function value $f(x_0)$ is deterministic).

Algorithm 1: **Minibatch Stochastic Three Points (`MiSTP`)**

Initialization

Choose $x_0 \in \mathbb{R}^d$, positive stepsizes $\{\alpha_k\}_{k\geq 0}$, probability distribution $\mathcal{D}$ on $\mathbb{R}^d$.

For $k = 0, 1, 2, \ldots$

1: Generate a random vector $s_k \sim \mathcal{D}$
2: Choose elements of the subset $\mathcal{B}_k$ u.a.r
3: Let $x_+ = x_k + \alpha_k s_k$ and $x_- = x_k - \alpha_k s_k$
4: $x_{k+1} = \arg\min\{f_{\mathcal{B}_k}(x_-), f_{\mathcal{B}_k}(x_+), f_{\mathcal{B}_k}(x_k)\}$

Lemma 1. *For $x \in \mathbb{R}^d$ such that x is independent from $\mathcal{B}$, i.e., the choice of x does not depend on the choice of $\mathcal{B}$, $f_{\mathcal{B}}(x)$ is an unbiased estimator of $f(x)$.*

Proof. See appendix A, section $A.1$. □

Throughout the paper, we assume that f_i, (for $i = 1, \ldots, n$) is differentiable, and has L_i-Lipschitz gradient. We assume also that f is bounded from below.

Assumption 1. *The objective function f_i, (for $i = 1, \ldots, n$) is L_i-smooth with $L_i > 0$ and f is bounded from below by $f_* \in \mathbb{R}$. That is, f_i has a Lipschitz continuous gradient with a Lipschitz constant L_i:*

$$\|\nabla f_i(x) - \nabla f_i(y)\|_2 \leq L_i\|x - y\|_2, \qquad \forall x, y \in \mathbb{R}^d$$

and $f(x) \geq f_*$ for all $x \in \mathbb{R}^d$.

Assumption 2. *We assume that the variance of $f_{\mathcal{B}}(x)$ is bounded for all $x \in \mathbb{R}^d$:*

$$\mathbf{E}_{\mathcal{B}}[(f(x) - f_{\mathcal{B}}(x))^2] < \sigma^2_{|\mathcal{B}|} < \infty$$

This assumption is very common in the stochastic optimization literature (Larson, Menickelly, and Wild 2019, section 6). Note that we put the subscript $|\mathcal{B}|$ in $\sigma_{|\mathcal{B}|}$ to mention that this deviation may be dependent on the minibatch size. Consider, for example, the case of sampling minibatches uniformly with replacement. In such case, the expected deviation between f and $f_{\mathcal{B}}$ satisfy $\mathbf{E}_{\mathcal{B}}[(f(x) - f_{\mathcal{B}}(x))^2] \leq \frac{A}{|\mathcal{B}|}$ for all $x \in \mathbb{R}^d$ independent from $\mathcal{B}$ where $A = \sup_{x\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^n (f_i(x) - f(x))^2$ (See appendix A, section $A.2$). Note that, given that the function $f(y) = y^2$ is convex on $\mathbb{R}$ and using Jensen's inequality we have: $(\mathbf{E}_{\mathcal{B}}[|f(x) - f_{\mathcal{B}}(x)|])^2 \leq \mathbf{E}_{\mathcal{B}}[(|f(x) - f_{\mathcal{B}}(x)|)^2]$. Therefore, $\mathbf{E}_{\mathcal{B}}[|f(x) - f_{\mathcal{B}}(x)|] \leq \sigma_{|\mathcal{B}|}$.

2.1 Assumption on the directions

Our analysis in the sequel of the paper will be based on the following key assumption.

Assumption 3. *The probability distribution $\mathcal{D}$ on $\mathbb{R}^d$ has the following properties:*

1. *The quantity $\mathbf{E}_{s\sim\mathcal{D}}\ \|s\|_2^2$ is positive and finite. Without loss of generality, in the rest of this paper we assume that it is equal to 1.*
2. *There is a constant $\mu_{\mathcal{D}} > 0$ and norm $\|\cdot\|_{\mathcal{D}}$ on $\mathbb{R}^d$ such that for all $g \in \mathbb{R}^d$,*

$$\mathbf{E}_{s\sim\mathcal{D}}\ |\langle g, s\rangle| \geq \mu_{\mathcal{D}}\|g\|_{\mathcal{D}}. \tag{3}$$

As proved in the `STP` paper (Bergou, Gorbunov, and Richtárik 2020), multiple distributions satisfy this assumption. For example: the uniform distribution on the unit sphere in $\mathbb{R}^d$ ($\mu_{\mathcal{D}} \sim \frac{1}{\sqrt{2\pi d}}$), the normal distribution with zero mean and $d \times d$ identity as the covariance matrix ($\mu_{\mathcal{D}} = \frac{\sqrt{2}}{\sqrt{\pi d}}$), the uniform distribution over standard unit basis vectors $\{e_1, ..., e_d\}$ in $\mathbb{R}^d$ ($\mu_{\mathcal{D}} = \frac{1}{d}$), the distribution on $S = s_1, ..., s_d$ where $\{s_1, ..., s_d\}$ form an orthonormal basis of $\mathbb{R}^d$ ($\mu_{\mathcal{D}} = 1$). We note that, unlike some other state-of-the-art methods like `RSGF` or `ZO-SVRG`, `MiSTP` does not rely on an approximation of the gradient using finite differences kind of techniques and allows multiple choices for the distribution $\mathcal{D}$ of the search directions.

2.2 Key lemma

Now, we establish the key result which will be used to prove the main properties of our algorithm.

Lemma 2. *If Assumptions 1, 2 , and 3 hold, then for all $k \geq 0$,*

$$\theta_{k+1} \leq \theta_k - \mu_{\mathcal{D}}\alpha_k g_k + \frac{L}{2}\alpha_k^2 + \sigma_{|\mathcal{B}|}, \tag{4}$$

where $L_{\mathcal{B}_k} = \frac{1}{|\mathcal{B}_k|}\sum_{i\in\mathcal{B}_k} L_i$, $L = \mathbf{E}[L_{\mathcal{B}_k}] = \frac{1}{n}\sum_{i=1}^n L_i$, $\theta_k = \mathbf{E}[f(x_k)]$ and $g_k = \mathbf{E}[\|\nabla f(x_k)\|_{\mathcal{D}}]$, and $|\mathcal{B}_k|$ is the minibatch size .

Proof. We have: $f(x_{k+1}) - f_{\mathcal{B}_k}(x_{k+1}) \leq |f(x_{k+1}) - f_{\mathcal{B}_k}(x_{k+1})|$ i.e.,

$$f(x_{k+1}) \leq f_{\mathcal{B}_k}(x_{k+1}) + |f(x_{k+1}) - f_{\mathcal{B}_k}(x_{k+1})| \tag{5}$$

We have: $x_{k+1} = \arg\min\{f_{\mathcal{B}_k}(x_k - \alpha_k s_k), f_{\mathcal{B}_k}(x_k + \alpha_k s_k), f_{\mathcal{B}_k}(x_k)\}$, therefore:

$$f_{\mathcal{B}_k}(x_{k+1}) \leq f_{\mathcal{B}_k}(x_k + \alpha_k s_k) \tag{6}$$

From L_i-smoothness of f_i we have:

$$f_i(x_k + \alpha_k s_k) \leq f_i(x_k) + \langle\nabla f_i(x_k), \alpha_k s_k\rangle + \frac{L_i}{2}\|\alpha_k s_k\|_2^2$$

By summing over f_i for $i \in \mathcal{B}_k$ and multiplying by $1/|\mathcal{B}_k|$ we get:

$$\begin{aligned} f_{\mathcal{B}_k}(x_k + \alpha_k s_k) &\leq f_{\mathcal{B}_k}(x_k) + \langle\nabla f_{\mathcal{B}_k}(x_k), \alpha_k s_k\rangle \\ &+ \frac{L_{\mathcal{B}_k}}{2}\|\alpha_k s_k\|_2^2 \\ &= f_{\mathcal{B}_k}(x_k) + \alpha_k\langle\nabla f_{\mathcal{B}_k}(x_k), s_k\rangle \\ &+ \frac{L_{\mathcal{B}_k}}{2}\alpha_k^2\|s_k\|_2^2 \end{aligned} \tag{7}$$

By using Inequalities (5), (6), and (7) we get:

$$\begin{aligned} f(x_{k+1}) \leq f_{\mathcal{B}_k}(x_k) + \alpha_k\langle\nabla f_{\mathcal{B}_k}(x_k), s_k\rangle \\ + \frac{L_{\mathcal{B}_k}}{2}\alpha_k^2\|s_k\|_2^2 + e^{k+1}_{\mathcal{B}_k} \end{aligned}$$

where $e^{k+1}_{\mathcal{B}_k} = |f(x_{k+1}) - f_{\mathcal{B}_k}(x_{k+1})|$

By taking the expectation conditioned on x_k and s_k and using Assumption 2 we get:

$$\mathbf{E}[f(x_{k+1})|x_k, s_k] \leq f(x_k) + \alpha_k \langle \nabla f(x_k), s_k \rangle + \frac{L}{2}\alpha_k^2 \|s_k\|_2^2 + \sigma_{|\mathcal{B}|}$$

Similarly, we can get (see details in appendix A, section $A.3$):

$$\mathbf{E}[f(x_{k+1})|x_k, s_k] \leq f(x_k) - \alpha_k \langle \nabla f(x_k), s_k \rangle + \frac{L}{2}\alpha_k^2 \|s_k\|_2^2 + \sigma_{|\mathcal{B}|}$$

From the two inequalities above we conclude:

$$\mathbf{E}[f(x_{k+1})|x_k, s_k] \leq f(x_k) - \alpha_k | \langle \nabla f(x_k), s_k \rangle | + \frac{L}{2}\alpha_k^2 \|s_k\|_2^2 + \sigma_{|\mathcal{B}|}$$

By taking the expectation over s_k and using Inequality (3) we get:

$$\mathbf{E}[f(x_{k+1})|x_k] \leq f(x_k) - \alpha_k \mu_{\mathcal{D}} \|\nabla f(x_k)\|_{\mathcal{D}} + \frac{L}{2}\alpha_k^2 + \sigma_{|\mathcal{B}|}$$

By taking expectation in the above inequality and due to the tower property of the expectation we get:

$$\mathbf{E}[f(x_{k+1})] \leq \mathbf{E}[f(x_k)] - \alpha_k \mu_{\mathcal{D}} \mathbf{E}[\|\nabla f(x_k)\|_{\mathcal{D}}] + \frac{L}{2}\alpha_k^2 + \sigma_{|\mathcal{B}|}$$

□

3 Complexity analysis

We first state, in Theorem 1, the most general complexity result of `MiSTP` where we do not make any additional assumptions on the objective functions besides smoothness of f_i, for $i = 1, \ldots, n$, and boundedness of f. The proofs follow the same reasoning as the ones in `STP` (Bergou, Gorbunov, and Richtárik 2020), we defer them to the appendix.

Theorem 1 (nonconvex case). *Let Assumptions 1, 2 , and 3 be satisfied and* $\sigma_{|\mathcal{B}|} < \frac{(\mu_{\mathcal{D}}\epsilon)^2}{2L}$. *Choose a fixed stepsize* $\alpha_k = \alpha$ *with* $(\mu_{\mathcal{D}}\epsilon - \sqrt{(\mu_{\mathcal{D}}\epsilon)^2 - 2L\sigma_{|\mathcal{B}|}})/L < \alpha < (\mu_{\mathcal{D}}\epsilon + \sqrt{(\mu_{\mathcal{D}}\epsilon)^2 - 2L\sigma_{|\mathcal{B}|}})/L$, *If*

$$K \geq k(\varepsilon) \overset{def}{=} \left\lceil \frac{f(x_0) - f_*}{\mu_{\mathcal{D}}\varepsilon\alpha - \frac{L}{2}\alpha^2 - \sigma_{|\mathcal{B}|}} \right\rceil - 1, \tag{8}$$

then $\min_{k=0,1,\ldots,K} \mathbf{E}\left[\|\nabla f(x_k)\|_{\mathcal{D}}\right] \leq \varepsilon$. *In particular, we have:* $\alpha_{optimal} = \mu_{\mathcal{D}}\varepsilon / L$

Proof. see appendix A, section $A.4$ □

Table 1 compares the convergence rate of our method `MiSTP` (when using the normal distribution or the uniform distribution on the unit sphere for the search directions) with two other state-of-the-art methods `RSGF` (Ghadimi and Lan 2013) and `ZO-SVRG` (Liu et al. 2018) in the non-convex case. `MiSTP` and `RSGF` have better dependence on d than `ZO-SVRG` ($\sqrt{d}$ vs. d) but worse dependence on K ($\frac{1}{\sqrt{K}}$ vs. $\frac{1}{K}$). In addition, `ZO-SVRG` suffers from an additional error term of order $O(\frac{1}{|\mathcal{B}|})$.

Table 1: Convergence rate for `MiSTP`, `RSGF`, and `ZO-SVRG` given K iterations.

Method	Convergence rate
MiSTP (This paper)	$O(\frac{\sqrt{d}}{\sqrt{K}})$
RSGF (Ghadimi and Lan 2013)	$O(\frac{\sqrt{d}}{\sqrt{K}})$
ZO-SVRG (Liu et al. 2018)	$O(\frac{d}{K} + \frac{1}{\lvert\mathcal{B}\rvert})$

We now state the complexity of `MiSTP` in the case of convex f. To do so, we add the following assumption:

Assumption 4. *We assume that f is convex, has a minimizer* x_*, *and has bounded level set at* x_0*:*

$$R_0 \overset{def}{=} \max\{\|x - x_*\|_{\mathcal{D}}^* \ : \ f(x) \leq f(x_0)\} < +\infty,$$

where $\|\xi\|_{\mathcal{D}}^* \overset{def}{=} \max\{\langle \xi, x\rangle \mid \|x\|_{\mathcal{D}} \leq 1\}$ *defines the dual norm to* $\|\cdot\|_{\mathcal{D}}$.

Note that if the above assumption holds, then whenever $f(x) \leq f(x_0)$, we get

$$\begin{aligned} f(x) - f(x_*) &\leq \langle \nabla f(x), x - x_* \rangle \\ &= \|\nabla f(x)\|_{\mathcal{D}} (x - x_*)^T \nabla f(x) / \|\nabla f(x)\|_{\mathcal{D}} \\ &\leq \|\nabla f(x)\|_{\mathcal{D}} \|x - x_*\|_{\mathcal{D}}^* \\ &\leq R_0 \|\nabla f(x)\|_{\mathcal{D}} \end{aligned}$$

That is,

$$\|\nabla f(x)\|_{\mathcal{D}} \geq \frac{f(x) - f(x_*)}{R_0}. \tag{9}$$

Theorem 2 (convex case). *Let Assumptions 1, 2 , 3, and 4 be satisfied. Let* $\varepsilon > 0$ *and* $\sigma_{|\mathcal{B}|} < \frac{(\mu_{\mathcal{D}}\epsilon)^2}{4LR_0^2}$, *choose constant stepsize* $\alpha_k = \alpha = \frac{\varepsilon\mu_{\mathcal{D}}}{LR_0}$, *If*

$$K \geq \frac{LR_0^2}{\mu_{\mathcal{D}}^2\varepsilon} \log\left(\frac{4(f(x_0) - f(x_*))}{\varepsilon}\right), \tag{10}$$

then $\mathbf{E}\left[f(x_K) - f(x_*)\right] \leq \varepsilon$.

Proof. see appendix A, section $A.5$ □

The theorems indicate that the condition on $\sigma_{|\mathcal{B}|}$ may imply that a larger minibatch size, $|\mathcal{B}|$, is necessary. This is particularly relevant when using biased gradient estimators like in `MiSTP`. Despite this, the numerical results in Section 4.1 demonstrate that small minibatch sizes can still be effective in practice. We note that many similar works in the literature dealing with biased gradient estimations require a larger minibatch size to ensure accurate theoretical results. This trend is exemplified, for instance, by these works: Bandeira, Scheinberg, and Vicente (2014); Bergou et al. (2022); Blanchet et al. (2016).

4 Numerical results

In this section, we report the results of some experiments conducted in order to evaluate the efficiency of `MiSTP`. All the presented results are averaged over 10 runs of the algorithms and the confidence intervals (the shaded region in the graphs) are given by $\mu \pm \frac{\sigma}{2}$ where μ is the mean and σ is the standard deviation. For each minibatch size, we choose the learning rate α by performing a grid search on the values 1,0.1,0.01,... and select the one that gives the best performance. τ denotes the minibatch size, i.e., $\tau = |\mathcal{B}|$. In all our implementations, the starting point x_0 is sampled from the standard Gaussian distribution. The distribution $\mathcal{D}$ used to sample search directions, unless specified otherwise, is the normal distribution with zero mean and $d \times d$ identity as the covariance matrix. [1]

4.1 MiSTP on ridge regression and regularized logistic regression problems

We performed experiments on ridge regression and regularized logistic regression. They are problems with strongly convex objective function f.

In the case of ridge regression we solve:

$$\min_{x \in \mathbb{R}^d} \left[f(x) = \frac{1}{2n} \sum_{i=1}^{n} (A[i,:]x - y_i)^2 + \frac{\lambda}{2} \|x\|_2^2 \right] \tag{11}$$

and in the case of regularized logistic regression we solve:

$$\min_{x \in \mathbb{R}^d} \left[f(x) = \frac{1}{2n} \sum_{i=1}^{n} \ln(1+\exp(-y_i A[i,:]x)) + \frac{\lambda}{2} \|x\|_2^2 \right] \tag{12}$$

In both problems $A \in \mathbb{R}^{n \times d}$, $y \in \mathbb{R}^n$ are the given data and $\lambda > 0$ is the regularization parameter. For logistic regression: $y \in \{-1, 1\}^n$ and all the values in the first column of A are equal to 1. [2] For both problems we set $\lambda = 1/n$. The experiments of this section are conducted using LIBSVM datasets (Chang and Lin 2011). We evaluate the performance of `MiSTP` when using different minibatch sizes and in comparison with some other state-of-the-art ZO methods. Additionally, in Appendix B, Section B.2, we report more experiments comparing the performance of `MiSTP` to `SGD`.

`MiSTP` with different minibatch sizes Figures 1 and 2 show the performance of `MiSTP` when using different minibatch sizes. From these figures, we see good performance of `MiSTP`. For different minibatch sizes, it generally converges faster than the original `STP` (the full batch) in terms of number of epochs. We notice also that there is an optimal minibatch size that gives the best performance for each dataset: among the tested values, for the 'abalone' dataset it is equal to 50, for 'splice' dataset it is 1, for 'a1a' and 'australian' datasets it is 10. All those optimal minibatch sizes are just a very small subset of the whole dataset which results in less computation at each iteration. Those results also show that we could get a good performance when using only an approximation of the objective function using a small subset of the data rather than the exact function evaluations.

[1] All codes for the experiments are available at: https://github.com/SoumiaBouch/Minibatch-STP.

[2] This is to account for the bias term. When using LIBSVM datasets we add this column to the data.

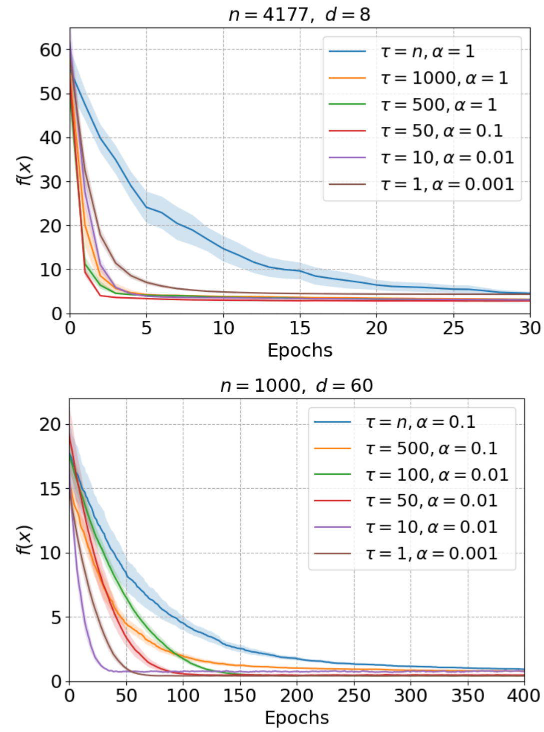

Figure 1: Performance of `MiSTP` with different minibatch sizes on ridge regression problem. Above, the abalone dataset. Below, the splice dataset.

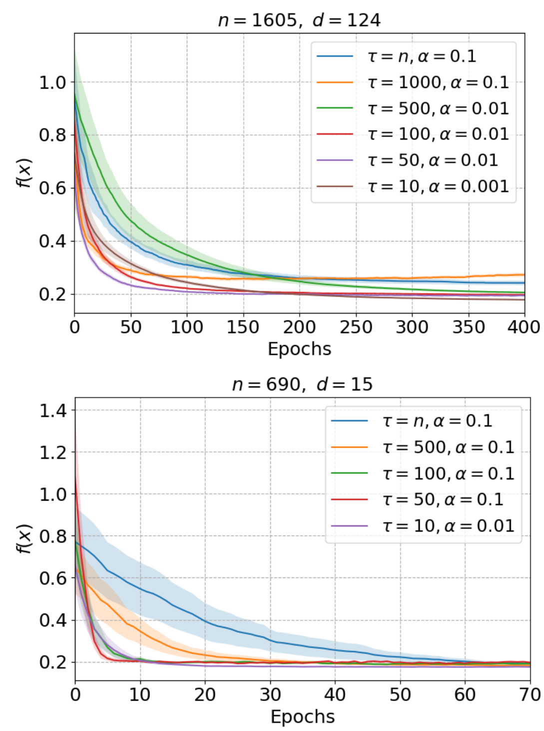

Figure 2: Performance of `MiSTP` with different minibatch sizes on regularized logistic regression problem. Above, the a1a dataset. Below, the australian dataset.

MiSTP vs. other zero-order methods In this section, we compare the performance of MiSTP with three other ZO optimization methods. The first is RSGF (Ghadimi and Lan 2013). In this method, at iteration k, the iterate is updated as follow:

$$x_{k+1} = x_k - \alpha_k \frac{f_{\mathcal{B}_k}(x_k + \mu_k s_k) - f_{\mathcal{B}_k}(x_k)}{\mu_k} s_k \qquad (13)$$

where $\mu_k \in (0, 1)$ is the finite differences parameter, α_k is the stepsize, s_k is a random vector following the uniform distribution on the unit sphere, and $\mathcal{B}_k$ is a randomly chosen minibatch. The second is ZO-SVRG(Liu et al. 2018, Algorithm 2). For this method, at iteration k, the gradient estimation of $f_{\mathcal{B}_k}$ at x_k is given by:

$$\hat{\nabla} f_{\mathcal{B}_k}(x_k) = \frac{d}{\mu}(f_{\mathcal{B}_k}(x_k + \mu s_k) - f_{\mathcal{B}_k}(x_k)) s_k \qquad (14)$$

where $\mu > 0$ is the smoothing parameter and s_k is a random direction drawn from the uniform distribution over the unit sphere. And the last is ZO-CD (ZO coordinates descent method), in this method, at iteration k, the iterate is updated as follow:

$$x_{k+1} = x_k - \alpha_k g_{\mathcal{B}_k},$$

$$\text{s.t.} \quad g_{\mathcal{B}_k} = \sum_{i=1}^{d} \frac{f_{\mathcal{B}_k}(x_k + \mu e_i) - f_{\mathcal{B}_k}(x_k - \mu e_i)}{2\mu} e_i \qquad (15)$$

where $\mu > 0$ is a smoothing parameter and $e_i \in \mathbb{R}^d$ for $i \in [d]$ is a standard basis vector with 1 at its ith coordinate and 0 elsewhere.

The distribution $\mathcal{D}$ used here for MiSTP is the uniform distribution on the unit sphere. For RSGF, ZO-SVRG, and ZO-CD, we chose $\mu_k = \mu = 10^{-4}$. Figures 3 and 4 show the objective function values against the number of function queries of the different ZO methods using different minibatch sizes. Note that one function query is the evaluation of one f_i for $i \in [n]$ at a given point. On the ridge regression problem, MiSTP, RSGF, and ZO-CD show competitive performance while ZO-SVRG needs much more function queries to converge. On the regularized logistic regression problem, MiSTP outperforms all the other methods. RSGF, ZO-CD, and ZO-SVRG need almost 5 times more function queries to converge than MiSTP for $\tau = 100$ and around 2 times more function queries than MiSTP for $\tau = 50$.

4.2 MiSTP in neural networks

Figure 5 shows the results of experiments using MiSTP as the optimizer in a multi-layer neural network (NN) for MNIST digit (LeCun et al. 1998) classification with different minibatch sizes. The architecture we used has three fully-connected layers of size 256, 128, 10, with ReLU activation after the first two layers and a Softmax activation function after the last layer. The loss function is the categorical cross entropy. From Figure 5 we observe that the minibatch size 6000 outperforms the minibatch size 3000 and the full batch, it converges faster to better accuracy and loss values. $\tau = 6000$ is $1/10$ of the dataset (we used the whole MNIST dataset which has 60000 samples), it leads to less computation time at each iteration than using all the 60000 samples. Besides it largely outperforms the full batch. Those results prove that minibatch training is more efficient

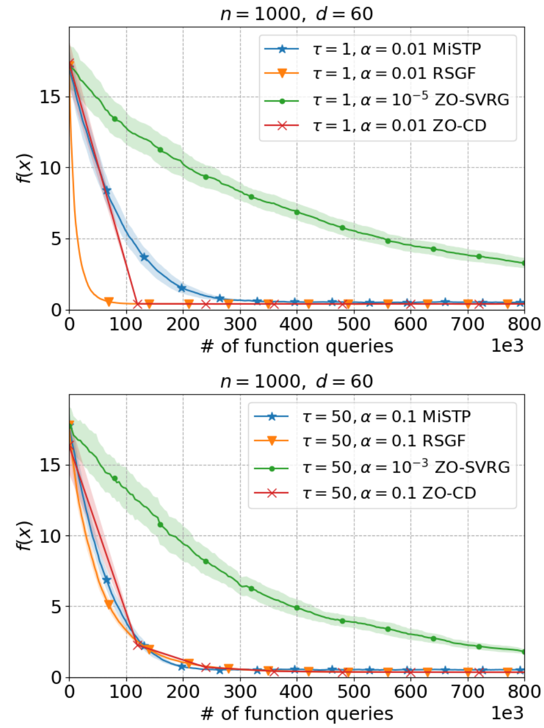

Figure 3: Comparison of MiSTP, RSGF, ZO-SVRG, and ZO-CD on the ridge regression problem using the splice dataset.

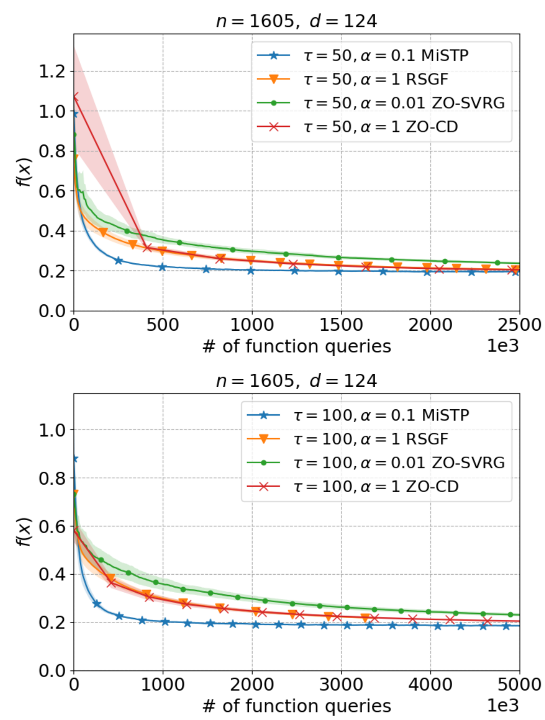

Figure 4: Comparison of MiSTP, RSGF, ZO-SVRG, and ZO-CD on the regularized logistic regression problem using the a1a dataset.

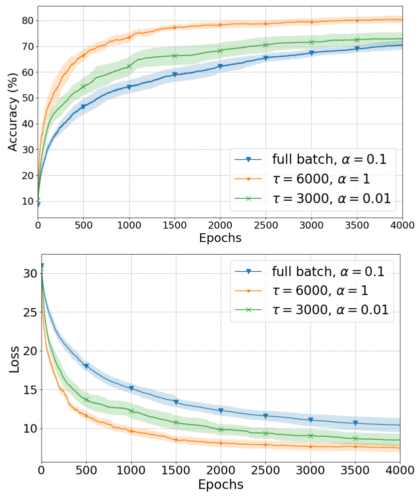

Figure 5: Comparison of different minibatch sizes for MiSTP in a multi-layer neural network.

than the full batch training and that we can find an optimal minibatch size that leads to efficient training of an NN in terms of performance and computation effort.

4.3 generation of black-box adversarial attacks on a deep neural network classifier

Existing studies have shown that well-trained deep neural networks (DNN) models can be vulnerable to adversarial examples (Papernot et al. 2017): given a benign input a whose label is initially correctly predicted by the model, it is possible to inject noise and craft an adversarial example a^{adv} that is almost indistinguishable from the original input but is mislabeled by the model with high confidence. Real-life DNN systems are black box systems. They do not release their internal architecture and weights; hence it is impossible to perform backpropagation to compute gradients and only the input and output of the targeted DNN are accessible. In this experiment, we generate a black box adversarial attack on a well-trained DNN classifier [3] for MNIST digit classification. The function f_i in (1) for this task is given by: $f_i(x) = \|a_i^{adv} - a_i\|_2^2 + c \cdot \max\{F_{y_i}(a_i^{adv}) - \max_{j \neq y_i} F_j(a_i^{adv}), 0\}$ s.t. $a_i^{adv} = 0.5 \cdot \tanh\left(\tanh^{-1} 2a_i + x\right)$. For $i = 1, ..., n$, a_i is the original benign image, y_i is its original class label, and a_i^{adv} is the generated adversarial image. The function $F(a)$ denotes the targeted classifier. It takes as input an image a and output a vector $F(a) \in [0,1]^{10}$ of confidence scores for each class. $c > 0$ is a regularization parameter. In f_i, the l_2 distortion loss $\|a_i^{adv} - a_i\|_2^2$ is used to enforce the similarities between the original image and the adversarial one. We compare the performance of our method MiSTP with three other methods RSGF (Ghadimi and Lan 2013), ZO-SVRG-Ave and ZO-SVRG (Liu et al. 2018). Following the same setup described in Liu et al. (2018), we generate an adversarial attack to a set of $n = 10$ images of class 1 using a minibatch size of $\tau = 5$ and a fixed stepsize $\alpha = 2$ for MiSTP, $\alpha = 5/d$ for ZO-SVRG, and $\alpha = 30/d$ for both RSGF and ZO-SVRG-Ave. We set the epoch length to 10, $\mu = 0.01$, and $c = 1$. The search directions for all the methods are sampled from the uniform distribution on the unit sphere. Figure 6 shows the total loss f versus number of epochs. MiSTP shows the best performance. It achieved lower loss values compared to the other methods. RSGF and ZO-SVRG-Ave have competitive performance while ZO-SVRG shows the worst performance. Table 2 shows the least l_2 distortion achieved by each method. MiSTP has the least score among all the methods which means that, compared to the other methods, MiSTP can produce adversarial examples that are less distinguishable from the original examples. The generated adversarial examples for all the methods are presented in Appendix B, Section B.1.

[3] https://github.com/carlini/nn_robust_attacks

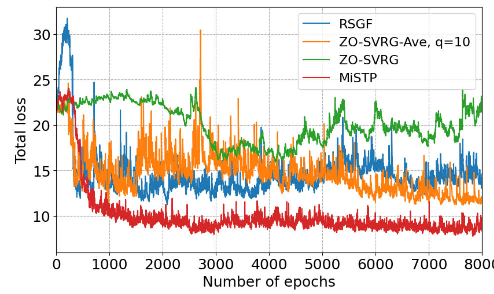

Figure 6: Performance of MiSTP, RSGF, ZO-SVRG, and ZO-SVRG-Ave for generation of black-box attack on a DNN classifier.

Table 2: Least l_2 distortion for each method

Method	$l_2$distortion
MiSTP	8.85
RSGF	11.24
ZO-SVRG-Ave, $q = 10$	11.15
ZO-SVRG	18.06

5 Conclusion

In this paper, we proposed the MiSTP method to extend the STP method to the case of using only an approximation of the objective function at each iteration assuming the error between the objective function and its approximation is bounded. MiSTP sample the search directions in the same way as STP, but instead of comparing the objective function at three points it compares an approximation. We derived our method's complexity in the case of nonconvex and convex objective function. The presented numerical results showed encouraging performance of MiSTP. In some settings, it showed superior performance over the original STP and some other ZO methods. There are a number of interesting future works to further extend our method, namely deriving a rule to find the optimal minibatch size, comparing the performance of MiSTP with other zero-order methods on deep neural networks problems, extending MiSTP to the case of distributed learning, and investigating MiSTP in the non-smooth case.

References

Al-Abri, S.; Lin, T. X.; Nelson, R. S.; and Zhang, F. 2021. A Derivative-free Distributed Optimization Algorithm with Applications in Multi-Agent Target Tracking. *2021 American Control Conference (ACC)*, 3844–3849.

Bandeira, A. S.; Scheinberg, K.; and Vicente, L. N. 2014. Convergence of trust-region methods based on probabilistic models. *SIAM Journal on Optimization*, 24(3): 1238–1264.

Bergou, E.; Diouane, Y.; Kungurtsev, V.; and Royer, C. W. 2022. A Stochastic Levenberg–Marquardt Method Using Random Models with Complexity Results. *SIAM/ASA Journal on Uncertainty Quantification*, 10(1): 507–536.

Bergou, E. H.; Gorbunov, E.; and Richtárik, P. 2020. Stochastic Three Points Method for Unconstrained Smooth Minimization. *SIAM Journal on Optimization*, 30(4): 2726–2749.

Bibi, A.; Bergou, E. H.; Sener, O.; Ghanem, B.; and Richtárik, P. 2020. A stochastic derivative-free optimization method with importance sampling: Theory and Learning to Control. *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 34(04): 3275–3282.

Blanchet, J.; Cartis, C.; Menickelly, M.; and Scheinberg, K. 2016. Convergence Rate Analysis of a Stochastic Trust Region Method for Nonconvex Optimization. *arXiv:1609.07428*.

Chang, C. C.; and Lin, C. J. 2011. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*.

Conn, A. R.; Scheinberg, K.; and Vicente, L. N. 2009. Introduction to Derivative Free Optimization. *Society for Industrial and Applied Mathematics (SIAM)*.

Conroy, B.; and Sajda, P. 2012. Fast, exact model selection and permutation testing for l2-regularized logistic regression. *Artificial Intelligence and Statistics*, 246–254.

Ghadimi, S.; and Lan, G. 2013. Stochastic First- and Zeroth-Order Methods for Nonconvex Stochastic Programming. *SIAM Journal on Optimization*, 23(4): 2341–2368.

Ghanbari, H.; and Scheinberg, K. 2017. Black-Box Optimization in Machine Learning with Trust Region Based Derivative Free Algorithm. *arXiv: 1703.06925*.

Golovin, D.; Solnik, B.; Moitra, S.; Kochanski, G.; Karro, J.; and Sculley, D. 2017. Google Vizier: A Service for Black-Box Optimization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, 1487–1495. New York, NY, USA: Association for Computing Machinery.

Gorbunov, E.; Bibi, A.; Sener, O.; Bergou, E. H.; and Richtárik, P. 2020. A stochastic derivative free optimization method with momentum. *International Conference on Learning Representations (ICLR)*.

Gower, R. M.; Loizou, N.; Qian, X.; Sailanbayev, A.; Shulgin, E.; and Richtárik, P. 2019. SGD: General Analysis and Improved Rates. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 5200–5209. PMLR.

Kramer, O.; Ciaurri, D. E.; and Koziel, S. 2011. *Derivative-Free Optimization*, 61–83. Computational Optimization, Methods and Algorithms. Springer.

Larson, J.; Menickelly, M.; and Wild, S. M. 2019. Derivative-free optimization methods. *Acta Numerica*, 28: 287–404.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Li, Y.; Tang, Y.; Zhang, R.; and Li, N. 2022. Distributed Reinforcement Learning for Decentralized Linear Quadratic Control: A Derivative-Free Policy Optimization Approach. *IEEE Transactions on Automatic Control*, 67(12): 6429–6444.

Liu, S.; Kailkhura, B.; Chen, P.-Y.; Ting, P.; Chang, S.; and Amini, L. 2018. Zeroth-order stochastic variance reduction for nonconvex optimization. *Advances in Neural Information Processing Systems (NeurIPS)*, 3731–3741.

Malik, D.; Pananjady, A.; Bhatia, K.; Khamaru, K.; Bartlett, P. L.; and Wainwright, M. J. 2020. Derivative-Free Methods for Policy Optimization: Guarantees for Linear Quadratic Systems. *Journal of Machine Learning Research*, 21(21): 1–51.

Nesterov, Y.; and Spokoiny, V. 2017. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17: 527–566.

Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z. B.; and Swami, A. 2017. Practical Black-Box Attacks against Machine Learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, 506–519. New York, NY, USA: Association for Computing Machinery.

P.Koch; Golovidov, O.; Gardner, S.; Wujek, B.; Griffin, J.; and Xu, Y. 2018. Autotune: A Derivative-Free Optimization Framework for Hyperparameter Tuning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, 443–452. New York, NY, USA: Association for Computing Machinery.

Reddi, S. J.; Hefny, A.; Sra, S.; Poczos, B.; and Smola, A. 2016. Stochastic variance reduction for nonconvex optimization. *International conference on machine learning*, 314–323.

Shen, X.; Alam, M.; Fikse, F.; and Rönnegård, L. 2013. A novel generalized ridge regression method for quantitative genetics. *Genetics*, 193(4): 1255–1268.

Suykens, J. A.; and Vandewalle, J. 1999. Least squares support vector machine classifiers. *Neural processing letters*, 9(3): 293–300.

Turner, R.; Eriksson, D.; McCourt, M.; Kiili, J.; Laaksonen, E.; Xu, Z.; and Guyon, I. 2021. Bayesian Optimization is Superior to Random Search for Machine Learning Hyperparameter Tuning: Analysis of the Black-Box Optimization Challenge 2020. In *Proceedings of the NeurIPS 2020 Competition and Demonstration Track*, volume 133 of *Proceedings of Machine Learning Research*, 3–26. PMLR.

Ughi, G.; Abrol, V.; and Tanner, J. 2021. An empirical study of derivative-free-optimization algorithms for targeted black-box attacks in deep neural networks. *Optimization and Engineering*.

Watanabe, S.; and Roux, J. L. 2014. Black box optimization for automatic speech recognition. *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference*, 3256–3260.