Sains Malaysiana 55(3)(2026): 491-501
http://doi.org/10.17576/jsm-2026-5503-11

A Mini-Batch Algorithm with Adaptive Learning Rate Strategy

(Algoritma Kelompok Mini dengan Strategi Kadar Pembelajaran Adaptif)

WEIJUAN SHI[1], ADIBAH SHUIB[2,*] & ZURAIDA ALWADOOD[2]

[1]*College of Mathematics and Finance, Hunan University of Humanities, Science and Technology, Loudi, China*
[2]*College of Computing, Informatics and Media (KPPIM), Universiti Teknologi MARA (UiTM), 40450 Shah Alam, Selangor, Malaysia*

Received: 31 December 2024/Accepted: 19 February 2026

ABSTRACT

To address the limitations of manually selecting step sizes or using diminishing step size sequences, which can slow convergence in mini-batch algorithms, we propose a strategy for automatically calculating step sizes by employing the Positive Defined Stabilized Barzilai-Borwein (PDSBB) method. The PDSBB step size is integrated into the mini-batch semi-stochastic gradient descent (mS2GD) algorithm, creating a novel algorithm called mS2GD-PDSBB. Based on the linear convergence result, the computational complexity is characterized in terms of the expected number of stochastic gradient evaluations required to achieve a prescribed accuracy level. Computational experiments on benchmark instances are conducted to evaluate the convergence behavior of the proposed algorithm. Suitable mini-batch size leads the mS2GD-PDSBB algorithm to successfully attain the performance consistent to the base algorithms. The numerical experiments demonstrate that the proposed mS2GD-PDSBB algorithm achieves stable and fast convergence with the adaptive step-size strategy. In particular, the algorithm shows reduced sensitivity to the choice of initial step sizes and consistently outperforms or matches mS2GD and mS2GD-BB in terms of objective sub-optimality and test error across different dataset.

Keywords: Adaptive step size; convergence rate; mS2GD algorithm

ABSTRAK

Untuk menangani batasan pemilihan saiz langkah secara manual atau penggunaan jujukan saiz langkah yang semakin berkurangan, yang boleh memperlahankan penumpuan dalam algoritma kelompok mini, kami mencadangkan strategi untuk mengira saiz langkah secara automatik dengan menggunakan kaedah *Positive Defined Stabilized Barzilai-Borwein* (PDSBB). Saiz langkah PDSBB disepadukan ke dalam algoritma penurunan kecerunan separa stokastik kelompok mini (mS2GD), mewujudkan algoritma baharu yang dipanggil mS2GD-PDSBB. Berdasarkan hasil penumpuan linear, kerumitan pengiraan dicirikan dari segi bilangan penilaian kecerunan stokastik yang dijangkakan yang diperlukan untuk mencapai tahap ketepatan yang ditetapkan. Uji kaji pengiraan pada contoh penanda aras dijalankan untuk menilai tingkah laku penumpuan algoritma yang dicadangkan. Saiz kelompok mini yang sesuai membawa algoritma mS2GD-PDSBB untuk berjaya mencapai prestasi yang tekal dengan algoritma asas. Uji kaji berangka menunjukkan bahawa algoritma mS2GD-PDSBB yang dicadangkan mencapai penumpuan yang stabil dan pantas dengan strategi saiz langkah adaptif. Khususnya, algoritma menunjukkan sensitiviti yang berkurangan terhadap pilihan saiz langkah awal dan secara tekal mengatasi atau memadankan mS2GD dan mS2GD-BB dari segi sub-keoptimuman objektif dan ralat ujian merentasi set data yang berbeza.

Kata kunci: Algoritma mS2GD; kadar penumpuan; saiz langkah adaptif

INTRODUCTION

Machine learning refers to a technique for analyzing data through automated construction of analytical models. This field is based on the concept that systems can derive insights from data, identify patterns, and make decisions with minimal human intervention (SAS 2003). The unconstrained optimization problem (1) is frequently utilized for binary classification in many machine learning scenarios.

$$\min_{w\in R^d} P(w) = F(w) + R(w) \tag{1}$$

where $F(w)$represents the average of numerous smooth functions $f_i(w)$, that is,

$$F(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w) \tag{2}$$

In this study, we choose each $f_i(w)$ is the logistic as in (3)

$$f_i(w) = \log\left(1+\exp(-b_i a_i^T w)\right) \tag{3}$$

where $(a_i, b_i) \in \mathbb{R}^d \times \{+1, -1\}$, $i = 1, \cdots, n$. Many active research topics, such as machine learning and (Khan et al. 2022); deep learning (Lin et al. 2022); object recognition and detection (Chen et al. 2022; Liu et al. 2020; Mittal, Ghosh & Singh 2024; Wen et al. 2023); compressed sensing and sparse reconstruction (Hosny, Shouman & Ali 2023; Nagahara 2024); as well as signal processing and convex optimization methods (Condat 2023), can be uniformly formulated as (2). While $R(w)$ is comparatively simple, yet it might be non-smooth and $w \in \mathbb{R}^d$ means that w is a d-dimensional real column vector. In this study, we choose

$$R(w) = \frac{\lambda_2}{2}\|w\|_2^2 + \lambda_1\|w\|_1 \tag{4}$$

where λ_1 and λ_2 are regularization parameters.

The traditional method used to tackle the problem outlined in (1) is the proximal gradient approach (Parikh & Boyd 2014; Xiao & Zhang 2014), characterized by its iterative structure described in Equation (5).

$$w_k = prox\ \eta_{kR}\left(w_{k-1} - \eta_k \nabla F\left(w_{k-1}\right)\right) \tag{5}$$

here, $\nabla F\left(w_{k-1}\right)$ represents the gradient of the objective function $\nabla F(w)$ evaluated at the point w_{k-1}, and $\eta_k > 0$ is the step size.

In many fields, step sizes have different names. For example, step size is known as learning rate in machine learning and it is called 'gain' in signal processing. 'Prox' denotes the proximity mapping or proximity operator, and it is defined as:

$$prox_R(z) \overset{def}{=} \arg\min_{w \in \mathbb{R}^d}\left\{\frac{1}{2}\|w - z\|^2 + R(w)\right\} \tag{6}$$

here, the symbol '$\overset{def}{=}$' denotes a definition, i.e., 'is defined as'. The traditional proximal gradient algorithm is limited to achieving a convergence rate of O $(1/k)$ in terms of objective sub-optimality, i.e., $P(w^k) - P(w_*) = O(1/k)$, where k denotes the number of iterations (Tseng 2000). The traditional proximal gradient algorithm is far from achieving the maximum descent rate O $(1/k^2)$ for convex objective functions.

Konečný et al. (2016) pointed out that when the accuracy reaches at ε, i.e., $P(w_*) - P(w) < \varepsilon$ (w_* = arg min P (w)), the minimum iteration steps required for the traditional proximal gradient algorithm is given as follows:

$$O\left(nL/\mu log\left(1/\varepsilon\right)\right) \tag{7}$$

here, L is a uniform upper bound on the Lipschitz constants of the gradients of f_i, μ is the strong convexity constant of P and $\varepsilon > 0$ is the prescribed accuracy.

Nesterov (2013) introduced Nesterov's accelerated proximal gradient algorithm. This enhancement boosts the convergence rate from O $(1/k)$ to O $(1/k^2)$. The Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) is a well-known and effective method, and its complexity for solving problem (1) was analyzed by Konečný et al. (2016) as follows:

$$O\left(n\sqrt{L/\mu}log\left(1/\varepsilon\right)\right) \tag{8}$$

Based on (5), the iterated format of Proximal Stochastic Gradient Method (Prox-SG) can be described as:

$$w_k = prox_{\eta_k R}\left(w_{k-1} - \eta_k \nabla f_i\left(w_{k-1}\right)\right) \tag{9}$$

where $\nabla f_i\left(w_{k-1}\right)$ is the gradient of the i-th component of the randomly selected objective function at w_{k-1}.

Mini-batching is a widely used acceleration technique for stochastic optimization problems. It effectively reduces the variance caused by random sampling and is simple to implement. Its suitability for parallel computation has made it an important research topic. Reddi et al. (2016) analyzed the convergence and complexity of Stochastic Variance Reduced Gradient (SVRG) in non-convex problems and introduced mini-batching to create the Mini-batch Stochastic Variance Reduced Gradient (MSVRG) method. Li et al. (2014) introduced a mini-batch update technique beyond traditional parameter averaging. Berahas, Nocedal and Takáč (2016) proposed mini-batch Limited-Memory Broyden-Fletcher-Goldfarb-Shanno Algorithm (L-BFGS). A mini-batch semi-stochastic gradient descent algorithm (mS2GD) was introduced by Konečný et al. (2016) for non-smooth objective functions. Yang et al. (2018) further enhanced this method by incorporating the Random BB method (RBB), resulting in the mS2GD-RBB algorithm. Additionally, they computed the step size for the Stochastic Recursive Gradient Algorithm (SARAH), first introduced by Nguyen et al. (2017), within the framework of mini-batch processing. This resulted in the development of a new variant of SARAH, referred to as the Mini-Batch Stochastic Recursive Gradient algorithm with Random Barzilai–Borwein step size (MB-SARAH-RBB) (Yang, Chen & Wang 2021). A momentum version, SARAH-M, further improve stability and speed (Yang 2024). Yang et al. (2019) integrated an online step size (OSS) into the MSVRG method, creating MSVRG-OSS. Ma et al. (2018) proposed Stabilized Barzilai-Borwein (SBB) step size, which incorporates an extra positive term into its denominator. Shao, Wang and Han (2022) combined BB with mirror descent and the Frank-Wolfe method, leading to MDBB and FWBB.

The earlier-mentioned have improved the shortcomings of the BB method, but they also have their own limitations. For example, both BB and RBB cannot continue when the denominator approaches zero. The SBB method does not

TABLE 1. Gap analysis about the variants of BB step size

No	Algorithm	Authors	Step Size	Denominator approaching zero	Sensitivity to initial step size	Sample size requirement
1	MSVRG-OSS	Yang et al. (2018)	OSS	No	Yes	Relatively high
2	mS2GD-BB	Yang et al. (2018)	BB	Yes	No	Minimal
3	mS2GD-RBB	Yang et al. (2018)	RBB	Yes	No	Minimal
4	SVRG-SBB	Ma et al. (2018)	SBB	No	No	Minimal
5	MB-SARAH-RBB	Yang, Chen & Wang (2021)	RBB	Yes	No	Minimal

tell us how to choose suitable parameter values, nor does it provide guidelines for parameter selection. In the MSVRG-OSS algorithm, the numerical performance is particularly influenced by both the mini-batch size and the initial step size. With a relatively large sample size, performance generally improves. However, this also leads to a heavier computational load and a slower convergence rate. Table 1 provides a summary of the gap analysis.

The structure of this paper is as follows: the first section introduces the research background, the second details the methodology, the third presents the analysis and discussion of the results, and the final section offers a summary.

MATERIALS AND METHODS

MS2GD-PDSBB ALGORITHM

Primarily applied to nonlinear optimization problems, the BB method is commonly referred to as the two-point step gradient method (Barzilai & Borwein 1988). The BB method fulfills the quasi-Newton property with very little computation, setting it apart from the traditional quasi-Newton approach. Some literature also equates satisfying the quasi-Newton property with meeting the secant equation. Consider the following optimization problem we need to solve:

$$\min f(w) \tag{10}$$

where $f(w)$ is differentiable.

The quasi-Newton method's iterative formula for problem (10) is expressed as Equation (11), where an estimate of the Hessian matrix at point w_t is denoted as B_t,

$$w_{t+1} = w_t - \mathrm{B}_t^{-1}\nabla f(w_t) \tag{11}$$

where $\nabla f(w_t)$ denotes the gradient of the objective function evaluated at w_t.

To approximate the Hessian matrix, use $B_t = \frac{1}{\eta_t} I$ ($\eta_t > 0$ is the step size) and substitute it into the secant equation $B_t\, s_t = y_t$, where $y_t = \nabla f(w_t) - \nabla f(w_{t-1})$, $t > 1$ and $s_t = w_t - w_{t-1}$ represent the changes in the gradient and the position, respectively. We then solve for the residual of the secant equation.

$$\min \left\| \tfrac{1}{\eta_t} s_t - y_t \right\|^2 \tag{12}$$

The BB step size can be expressed as:

$$\eta_t^{BB1} = \frac{\|s_t\|^2}{s_t^T y_t} \tag{13}$$

Barzilai and Borwein proposed a method for deriving another BB step size

$$\eta_t^{BB2} = \frac{s_t^T y_t}{\|y_t\|^2} \tag{14}$$

which is obtained by solving the minimization problem (14)

$$\min \left\| s_t - \eta_t y_t \right\|^2 \tag{15}$$

The first-order algorithms increasing the risk of getting stuck in local minimum when the step size is too small. Conversely, a too large step size can lead to zig-zagging or oscillating, likely preventing the algorithm from reaching the optimal solution.

The step size in the BB method is determined by assessing the difference between two consecutive gradient vectors. The denominator gets close to zero when this difference becomes very small. Such a situation may lead the optimization algorithm to instability or divergence. To address this issue, Shi, Shuib and Alwadood (2023) proposed a solution. Modifications were made to the initial BB step size, with the details outlined as follows:

Given a predefined threshold $\varepsilon > 0$, if $s_t^T y_t < \varepsilon$, η_t is computed according to the BB step size in (13); otherwise, η_t is assigned as the mean of the previous step sizes. That is

$$\eta_t = \begin{cases} \eta_t^{BB1}, s_t^T y_t > \varepsilon \\ \sum_{i=1}^{t-1} \eta_i, s_t^T y_t < \varepsilon \end{cases} \tag{16}$$

It can be seen that Equation (16) is closely related to the classical BB step size, as part of the formula remains the same. The difference only arises when $s_t^T y_t < \varepsilon$ is satisfied. In this case, the denominator of the BB step size approaches zero or becomes negative, and the proposed step size is therefore set to $\eta_t = \sum_{i=1}^{t-1} \eta_i$. This approach adjusts the step size automatically, keeping it stabilized at a suitable value. It is then integrated with the mS2GD algorithm, resulting in a new algorithm called mS2GD-PDSBB, as shown herewith:

Algorithm 1: mS2GD with PDSBB step size (mS2GD-PDSBB)

1: Input: m, initial step size η_0, initial point $\tilde{w}_0$, small positive parameter ε, mini-batch size $b \in [n]$, where n is the total number of samples.

2: For $k = 0,1, \cdots$ do

3: Set $\tilde{w} = \tilde{w}_k$

4: Computer and store

$$g_k = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\tilde{w}), g_k' = \partial P(\tilde{w})$$

5: if $k > 0$, then

$$\gamma_k = \frac{b}{m} \| \tilde{w}_k - \tilde{w}_{k-1} \|_2^2 / (\tilde{w}_k - \tilde{w}_{k-1})^T (g_k' - g_{k-1}')$$

6: if $(\tilde{w}_k - \tilde{w}_{k-1})^T (g_k' - g_{k-1}') > \varepsilon$

$$\eta_k = \gamma_k$$

7: else

$$\eta_k = \frac{1}{k} \sum_{i=0}^{k-1} \eta_i$$

8: end if

9: Set $w_0 = \tilde{w}$

10: Randomly choose $t_k \in \{1, \cdots, m\}$

11: for $t = 0, 1, \cdots, t_k - 1$ do

12: Randomly choose mini-batch $I_t \subset \{1, \cdots, n\}$ of size b

13: Computer a stochastic estimate of $\nabla F(w_t)$

14: $G_t = \nabla \Phi_{I_t}(w_t) - \nabla \Phi_{I_t}(\tilde{w}) + g_k$

15: $w_{t+1} = prox_{\eta_k R}(w_t - \eta_k G_t)$

16: end for

17: $\tilde{w}_{k+1} = w_{t_k}$

18: end for

To help the reader better understand Algorithm 1, several explanatory remarks are provided.

a) To address nonsmooth optimization problems, the PDSBB step size is computed using subgradients of the objective function

b)When the PDSBB step size is chosen to be fixed, the proposed algorithm reduces to the mS2GD algorithmc

c)An additional scaling factor $\frac{b}{m}$ is introduced in the step size computation. Here, b denotes the number of randomly sampled data points, and m represents the maximum number of inner iterations. This factor plays a crucial role in controlling the convergence behavior, as evidenced by the theoretical analysis.

Assumption 1 Each component function $f_i(w)$: $\mathbb{R}^d \to \mathbb{R}$ is differentiable and has an L-Lipschitz continuous gradient, i.e., there exists a constant $L_i > 0$ such that for any $w, v \in \mathbb{R}^d$

$$\|\nabla f_i(w) - f_i(v)\|_2 \le L_i \|w - v\|_2$$

Consequently, the full objective function $F(w)$ also has an L-Lipschitz continuous gradient, that is,

$$\|\nabla F(w) - F(v)\|_2 \le L \|w - v\|_2, \forall w, v \in \mathbb{R}^d$$

Assumption 2 The function $P(w)$ is μ-strongly convex, meaning there exists a constant $\mu > 0$ such that for all $w, v \in \mathbb{R}^d$, the following inequality is satisfied:

$$P(w) \ge P(v) + \xi^T (w - v) + \frac{\mu}{2} \|w - v\|_2^2, \forall \xi \in \partial P(w)$$

where $\partial P(w)$ denotes sub-gradient of the objective function at point w.

Lemma 1 Let $R(w)$ be a closed convex function on $\mathbb{R}^d$, and let $w, v \in \mathbb{R}^d$. Then, the following inequality holds for the proximal operator:

$$\| prox_R(w) - prox_R(v) \| \le \|w - v\|$$

where $prox_R$ denotes the proximal operator associated with R.

Lemma 2 Let $\alpha(b) \overset{def}{=} \left(\frac{n-b}{b(n-1)} \right)$. Based on this definition, consider G_t as defined in Algorithm 1. Then $E[G_t] = \nabla F(w_t)$, and the following variance holds:

$$E\| G_t - \nabla F(w_t) \|^2 \le 4L\alpha(b) \left[P(w_t) - P(w_*) + P(\tilde{w}) - P(w_*) \right]$$

Theorem 1 Based on Assumption 1, Assumption 2, Lemma 1 and Lemma 2, let $w_* \overset{def}{=} \arg\min_w P(w)$, and choose $b \in \{1, \cdots, n\}$. Additionally, assume that $\frac{4L\alpha(b)b}{m\mu} < 1$, and m is sufficiently large. Under these conditions, the following holds:

$$\rho \overset{def}{=} \frac{1}{b\left[1 - \frac{4L\alpha(b)b}{m\mu}\right]} + \frac{4L\alpha(b)b(m+1)}{m\left[\mu m - 4L\alpha(b)b\right]} < 1$$

where $a(b) = ((n-b)/b(n-1))$. Here, $P(w)$ denotes the objective function under consideration, and $w \in \mathbb{R}^d$ represents the decision variable. The notation $\arg\min_w P(w)$ denotes the set of minimizers of $P(w)$, and w_* refers to an optimal solution of the optimization problem. Subsequently, mS2GD-PDSBB exhibits linear convergence on expectation, characterized by a rate denoted as ρ:

$$E\left[P(w_k) - P(w_*)\right] \le \rho^k \left[P(w_0) - P(w_*)\right]$$

Proof The proof follows the methodology outlined in Xiao and Zhang (2014). For the sake of simplicity, the stochastic mapping is defined as follows:

$$d_t := \frac{1}{\eta_k}(w_t - w_{t+1}) = \frac{1}{\eta_k}\left(w_t - prox_{\eta_k R}(w_t - \eta_k G_t)\right)$$

This results in the update iterations being expressed as $w_{t+1} = w_t - \eta_k d_t$, let estimate

$$\begin{aligned} &\|w_{t+1} - w_*\|^2 \\ &= \|w_t - \eta_k d_t - w_*\|^2 \\ &= \|w_t - w_*\|^2 - 2\eta_k d_t^T (w_t - w_*) + \eta_k^2 \|d_t\|^2 \end{aligned} \tag{17}$$

Applying Lemma 3.7 in the reference (Xiao & Zhang 2014) with $w = w_t$, $v = G_t$, $w^+ = w_{t+1}$, $g = d_t$, $y = w_*$ and $\Delta = \Delta_t = G_t - \nabla F(w_t)$, we have

$$\begin{aligned} &-d_t^T (w_t - w_*) + \frac{\eta_k}{2}\|d_t\|^2 \\ &\le P(w_*) - P(w_{t+1}) - \frac{\mu_F}{2}\|w_t - w_*\|^2 - \frac{\mu_R}{2}\|w_{t+1} - w_*\|^2 - \Delta_t^T (w_{t+1} - w_*) \end{aligned} \tag{18}$$

Combine (17) and (18), we have:

$$\|w_{t+1} - w_*\|^2 \le \|w_t - w_*\|^2 - 2\eta_k\left[P(w_{t+1}) - P(w_*)\right] - 2\eta_k \Delta_t^T (w_{t+1} - w_*) \tag{19}$$

To estimate $-2\Delta_t^T(w_{t+1} - w_*)$, we define $\bar{w}_{t+1} = prox_{\eta_k R}(w_t - \eta_k \nabla F(w_t))$, then

$$\begin{aligned} &-2\eta_k \Delta_t^T (w_{t+1} - w_*) \\ &= -2\eta_k \Delta_t^T (w_{t+1} - \bar{w}_{t+1}) - 2\eta_k \Delta_t^T (\bar{w}_{t+1} - w_*) \\ &\le 2\eta_k \|\Delta_t\| \|w_{t+1} - \bar{w}_{t+1}\|^2 - 2\eta_k \Delta_t^T (\bar{w}_{t+1} - w_*) \\ &= 2\eta_k \|\Delta_t\| \left\|prox_{\eta_k R}(w_t - \eta_k G_t) - prox_{\eta_k R}(w_t - \eta_k \nabla F(w_t))\right\| - 2\eta_k \Delta_t^T (\bar{w}_{t+1} - w_*) \\ &\le 2\eta_k^2 \|\Delta_t\|^2 - 2\eta_k \Delta_t^T (\bar{w}_{t+1} - w_*) \end{aligned}$$

Combine with (19), we obtain

$$\|w_{t+1} - w_*\|^2 \le \|w_t - w_*\|^2 - 2\eta_k\left[P(w_{t+1}) - P(w_*)\right] + 2\eta_k^2 \|\Delta_t\|^2 - 2\eta_k \Delta_t^T (\bar{w}_{t+1} - w_*)$$

Take expectation on both side on the inequality above, we get:

$$E\|w_{t+1} - w_*\|^2 \le \|w_t - w_*\|^2 - 2\eta_k\left[EP(w_{t+1}) - P(w_*)\right] + 2\eta_k^2 E\|\Delta_t\|^2 - 2\eta_k E\left[\Delta_t^T (\bar{w}_{t+1} - w_*)\right]$$

Note that $E[\Delta_t] = 0$, combine with Lemma 2, we get

$$\begin{aligned} &E\|w_{t+1} - w_*\|^2 \\ &\le \|w_t - w_*\|^2 - 2\eta_k\left[EP(w_{t+1}) - P(w_*)\right] + 8L\alpha(b)\eta_k^2\left[P(w_t) - P(w_*) + P(\tilde{w}) - P(w_*)\right] \end{aligned}$$

We choose $\eta_k = \frac{b}{m}\frac{\|s_k\|^2}{s_k^T y_k}$, or $\eta_k = \frac{b}{m}\frac{1}{k}\sum_{i=0}^{k-1}\eta_i$. By leveraging the strong convexity of $F(w)$, it becomes straightforward to have the upper bound:

$$\begin{aligned} \eta_k &= \frac{b}{m} \cdot \frac{\|\tilde{w}_k - \tilde{w}_{k-1}\|^2}{(\tilde{w}_k - \tilde{w}_{k-1})^T (g_k' - g_{k-1}')} \\ &\le \frac{b}{m} \cdot \frac{\|\tilde{w}_k - \tilde{w}_{k-1}\|^2}{\mu\|\tilde{w}_k - \tilde{w}_{k-1}\|^2} \\ &= \frac{b}{m\mu}. \end{aligned}$$

Or $\eta_k = \frac{1}{k}\sum_{i=0}^{k-1}\eta_i$, according to the choosing of every η_i, we have $\eta_k \le \frac{1}{k} \cdot k \cdot \frac{b}{m\mu} = \frac{b}{m\mu}$.

As shown in the above proof $\eta_k \le \frac{b}{m\mu}$, it follows that the inequality $0 < \eta_k \le \frac{1}{L}$ holds when the parameter m is sufficiently large, as required in Xiao and Zhang (2014). Then

$$\begin{aligned} &E\|w_{t+1} - w_*\|^2 \\ &\le \|w_t - w_*\|^2 - 2\frac{b}{m\mu}\left[EP(w_{t+1}) - P(w_*)\right] + 8L\alpha(b)\frac{b^2}{m^2\mu^2}\left[P(w_t) - P(w_*) + P(\tilde{w}) - P(w_*)\right] \end{aligned}$$

For a fixed k, it holds that $w_0 = \tilde{w} = \tilde{w}_k$. According to the definition of $\tilde{w}_{k+1}$, we obtain

$$E\left[P(\tilde{w}_{k+1})\right] = \frac{1}{m}\sum_{t=1}^{m} E\left[P(w_t)\right]. \tag{20}$$

Sum the inequality for $1 \le t \le m$, we obtain

$$E\|w_{t+1} - w_*\|^2 + 2\frac{b}{m\mu}\left[EP(w_m) - P(w_*)\right] + 2\frac{b}{m\mu}\left(1 - 4L\alpha(b)\frac{b}{m\mu}\right)\sum_{t=1}^{m-1}\left[EP(w_t) - P(w_*)\right]$$

$$\le \|w_0 - w_*\|^2 + 8L\alpha(b)\frac{b^2}{m^2\mu^2}\left[P(w_0) - P(w_*) + m\left(P(\tilde{w}) - P(w_*)\right)\right]$$

Further,

$$\frac{2b}{m\mu}\left(1 - 4L\alpha(b)\frac{b}{m\mu}\right)\sum_{t=1}^{m}\left[EP(w_t) - P(w_*)\right] \le \|\tilde{w} - w_*\|^2 + 8L\alpha(b)(m+1)\frac{b^2}{m^2\mu^2}\left[P(\tilde{w}) - P(w_*)\right]$$

The convexity of P implies $\|\tilde{w} - w_*\|^2 \le \frac{2}{\mu}P(\tilde{w}) - P(w_*)$, by using (20), we have

$$\frac{2b}{m\mu}\left(1-4L\alpha(b)\frac{b}{m\mu}\right)m\left[EP(\tilde{w}_{k+1})-P(w_*)\right]\le\left(\frac{2}{\mu}+8L\alpha(b)(m+1)\frac{b^2}{m^2\mu^2}\right)\left[P(\tilde{w}_k)-P(w_*)\right]$$

That is,

$$\begin{aligned}&EP(\tilde{w}_{k+1})-P(w_*)\\&\le\left(\frac{1}{b\left[1-\frac{4L\alpha(b)b}{m\mu}\right]}+\frac{4L\alpha(b)b(m+1)}{m\left[\mu m-4L\alpha(b)b\right]}\right)[P(\tilde{w}_k)-P(w_*)]\\&=\rho[P(\tilde{w}_k)-P(w_*)]\end{aligned}$$

By using the inequality mentioned above, we finalize the proof of Theorem 1.

COMPLEXITY ANALYSIS

In this section, we analyze the computational complexity of the proposed mS2GD-PDSBB algorithm. Under the standard smoothness and strong convexity assumptions, Theorem 1 establishes a linear convergence rate in expectation with respect to the objective sub-optimality. Based on this result, we further derive the iteration complexity and gradient complexity required to achieve an ε-accurate solution.

When $m \gg 1$, based on Theorem 1, we have

$$\rho\approx\frac{1}{b\left[1-\frac{4L\alpha(b)b}{m\mu}\right]}+\frac{4L\alpha(b)b}{\mu m-4L\alpha(b)b}, \text{ set } m=\frac{9L\alpha(b)b}{\mu}, \text{ i.e. } m$$

$= 9a(b)b_k$, then $\rho\approx\frac{9}{5b}+\frac{4}{5}$.

This analysis indicates that choosing $b \ge 9$ is sufficient to ensure the desired inequality. In practical implementations, however, mini-batch sizes are commonly selected as $b = 1, 2, 4, 8, 16\cdots$ among which $b \ge 16$ meets the theoretical requirement.

In particular, if let $s=\lceil\log(1/\varepsilon)\rceil$, let $m = 9a(b)b_k$, then need $(n + 9a(b)b_k)\log(1/\varepsilon)$ iterations for mS2GD-PDSBB achieves ε-accuracy solution.

RESULTS AND DISCUSSION

In this section, the performance of the mS2GD-PDSBB algorithm is assessed by applying it to a widely recognized machine learning benchmark problem: Logistic Regression (LR) with l_2-norm regularization:

$$\min_{w\in R^d}P(w)=\frac{1}{n}\sum_{i=1}^{n}\log\left(1+\exp\left(-b_i a_i^T w\right)\right)+\frac{\lambda_2}{2}\|w\|_2^2+\lambda_1\|w\|_1$$

where a_i represents the feature vector, while $b_i\in\{+1,-1\}$ corresponds to the class label for the i-th data point, $i = 1, \cdots, n$. The parameters λ_1 and λ_2, both strictly positive. To perform the numerical experiment, three standard real datasets, namely cod-rna, ijcnn1, and w8a obtained from LIBSVM are used. Its official website can be accessed at: http://www.csie.ntu.edu.tw/~cjlin/libsvm. Table 2 provides detailed information about the datasets as well as the parameters λ_1, λ_2.

During the initial stage, the sub-optimality and error rates were assessed across the three datasets, with varying sample sizes. In Figure 1, the Number of Effective Passes is represented on the horizontal axis. An Effective Pass is defined as the computation of the gradient of n f_i function. The sub-optimality, represented as $P(w) - P(w_*)$, is shown on the vertical axis, where w_* refers to the optimal solution obtained by applying mS2GD with the best tuned step size. The sample size b is selected from the set {1,2,4,8,16,32} while the initial step size η is fixed at 0.1.

In Figure 1, the numerical performance of mS2GD-PDSBB improves when the sample sizes b are 2, 4, 8, and 16, compared to the case where $b = 1$. However, this does not indicate that increasing the sample size always results in improved numerical performance. This can be seen from Figure 1. Conversely, using a larger sample size might significantly escalate the algorithm's computational burden.

In Figure 2, the step size dynamics of the mS2GD-PDSBB algorithm are shown for the three datasets, where initial step sizes of $\eta_0 = 0.1$, 1, and 10 are applied for each dataset, respectively. The number of Effective Passes is shown on the horizontal axis, while the vertical axis depicts the step size across the different datasets. It can be observed that after about 15 epochs, the step size stabilizes to a constant value across different datasets, implying that the choice of initial step size has little impact on the algorithm.

Figure 3 presents the test error rate. The results are presented for different initial step sizes $\eta_0 = 0.1$, 1, and 10. As shown in Figure 3, within a limited number of epochs, a relatively low error rate is achieved by the mS2GD-PDSBB algorithm across all three datasets.

In the second phase, the performance of mS2GD-PDSBB is highlighted by evaluating its sub-optimality, step size, and test error rate in comparison to the mS2GD

TABLE 2. Information regarding the data and parameters used in the experiments

Datasets	n	d	λ_1	λ_2
cod-rna	59535	8	10^{-5}	10^{-4}
w8a	49,749	300	10^{-5}	10^{-4}
ijcnn1	49,990	22	10^{-5}	10^{-4}

Note: n denotes the number of samples, while d indicates the data dimension

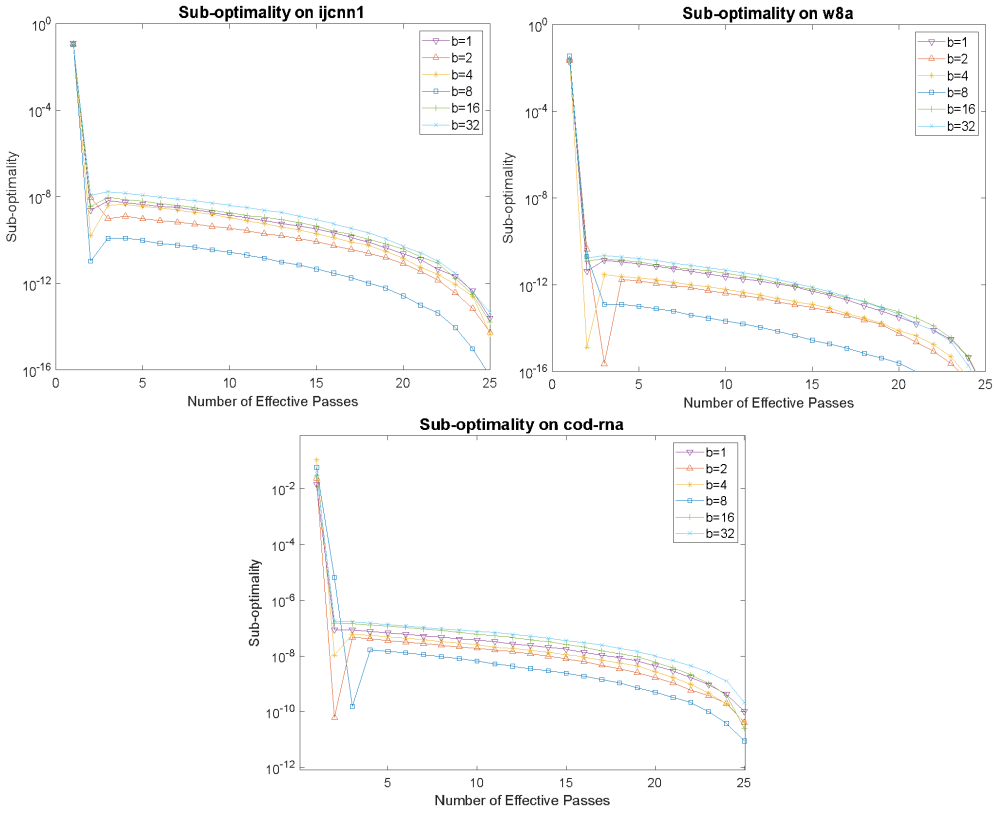

FIGURE 1. The sub-optimality performance of algorithm mS2GD-PDSBB with different sample number on ijcnn1, w8a, and cod-rna, respectively

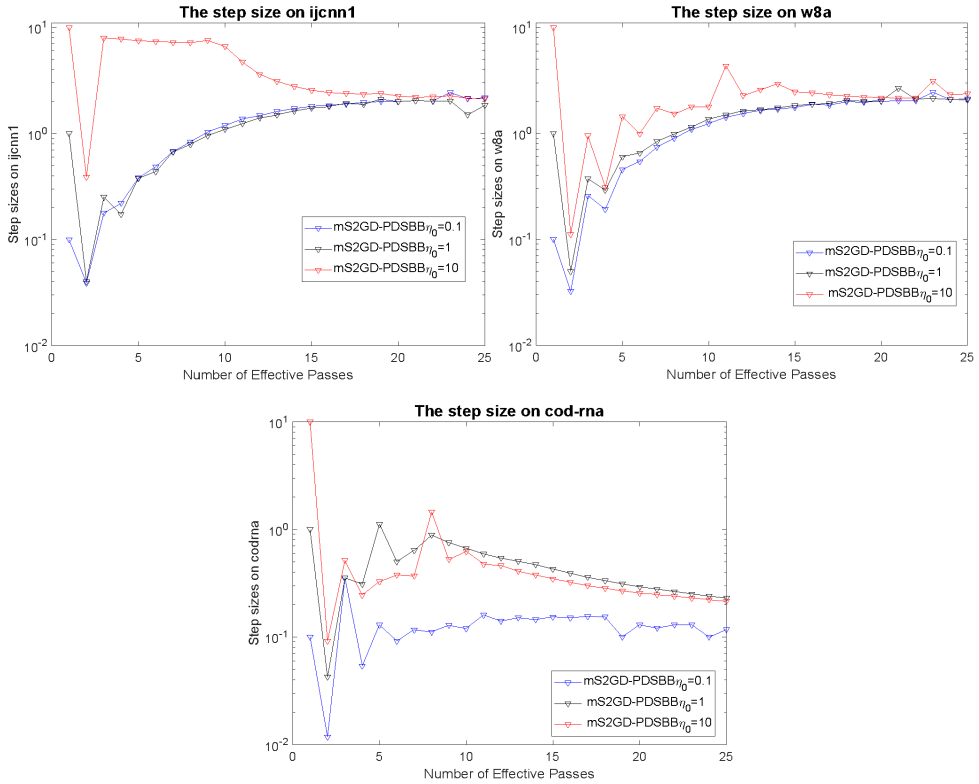

FIGURE 2. The step size of algorithm mS2GD-PDSBB with different initial step size on ijcnn1, w8a, and cod-rna, respectively

and mS2GD-BB across the three datasets. A fixed step size is selected for each dataset and tuned to its best performance for mS2GD. To showcase the insensitivity of the mS2GD-PDSBB algorithm to the choice of initial step size, three different values (η_0 = 0.1, 1, or 10) are assigned. Parameter b remains fixed at 8 for both mS2GD-PDSBB and mS2GD-BB.

In Figure 4, the step size is plotted on the y-axis, while the Number of Effective Passes axis is presented by the x-axis. The three sub-figures illustrate the results for mS2GD, mS2GD-BB, and mS2GD-PDSBB across various datasets, each with varying initial step sizes. Each sub-figure contains detailed information. For example, in the first sub-figure for the ijcnn1 dataset, solid lines with asterisks, squares, and triangles denote the step size evolution of mS2GD, mS2GD-BB, and mS2GD-PDSBB, respectively, each evaluated with initial step sizes of 0.1, 1, and 10.

Under various iterations, Figure 5 highlights how mS2GD-BB, mS2GD, and mS2GD-PDSBB compare in terms of sub-optimality performance. As previously mentioned, the sub-optimality, represented as $P(w) - P(w_*)$, is shown on the vertical axis, while the x-axis represents the Number of Effective Passes across different datasets.

The three sub-graphs clearly demonstrate that suboptimality levels have surpassed 10^{-12} on every dataset. The mS2GD-PDSBB algorithm effectively attains sub-optimality comparable to that achieved by both mS2GD and mS2GD-BB. For dataset cod-rna, mS2GD (η = 10) outer performs the sub-optimality for all of the algorithms. Meanwhile, mS2GD-PDSBB surpass the performance of all other algorithms starting from Number of Effective Passes equal to 5. From the graphs, it is evident that variations in the initial step size have little impact on sub-optimality, indicating that the proposed mS2GD-PDSBB algorithm is robust to different initial choices. While the original mS2GD can reach a comparable level of optimization with carefully tuned step sizes, its performance is highly sensitive to this tuning. In contrast, our method maintains consistently stable performance without requiring extensive step-size adjustment.

Presented in Figure 6, the error rate for the algorithms mS2GD-BB, mS2GD, and mS2GD-PDSBB across various datasets is demonstrated. It can be seen from these subplots that the error rate performance of the algorithms is generally satisfactory. For both the ijcnn1 and w8a datasets, the error rates quickly drop below 10% within a few iterations and eventually reach a stable level.

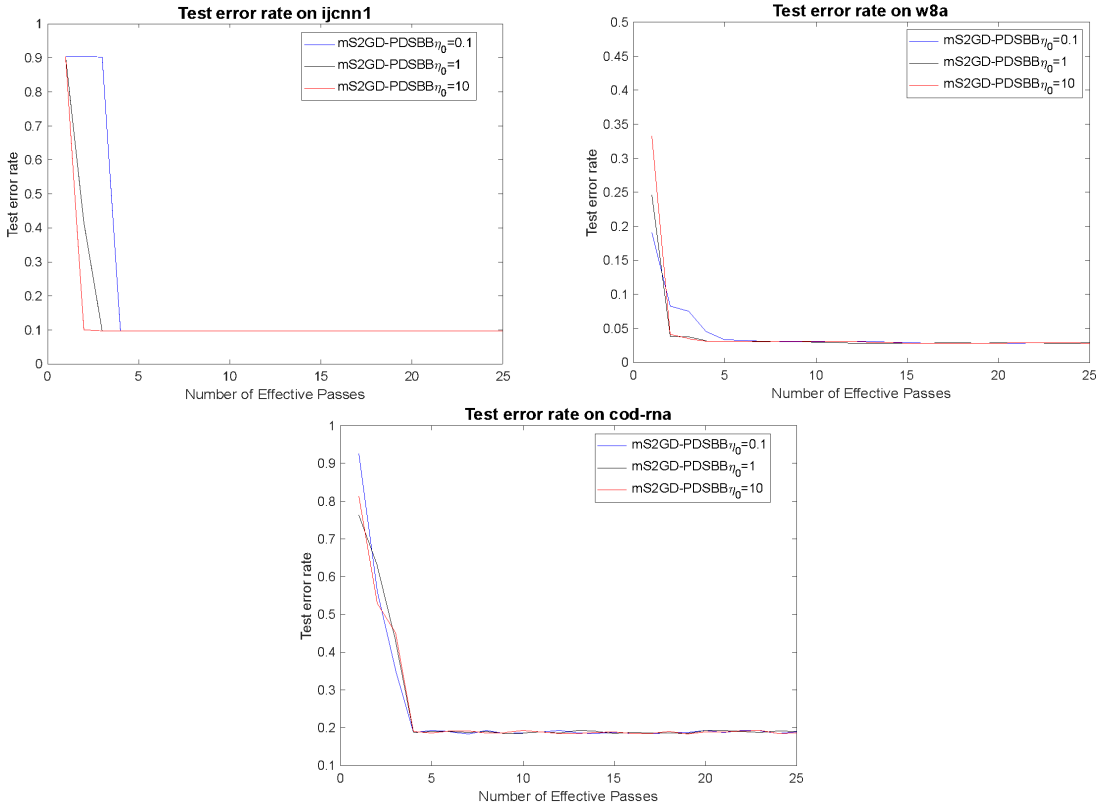

FIGURE 3. The test error rate of algorithm mS2GD-PDSBB with different initial step size on ijcnn1, w8a, and cod-rna, respectively

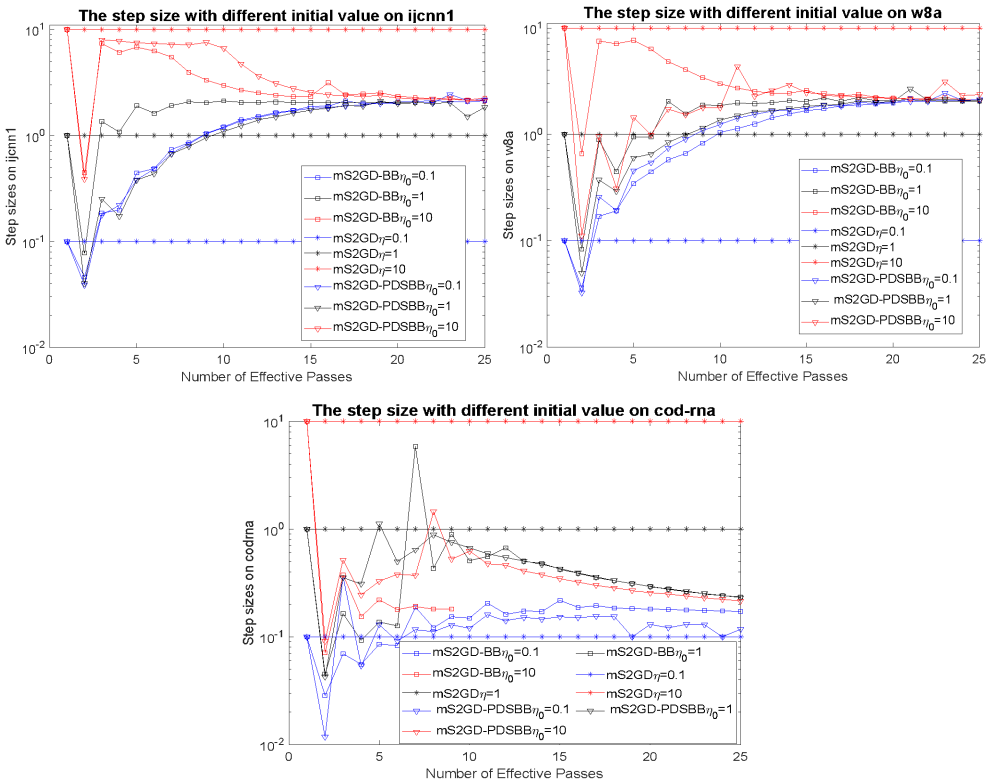

FIGURE 4. The step size for mS2GD-BB, mS2GD, and mS2GD-PDSBB varies with different initial step sizes on the ijcnn1, w8a, and cod-rna datasets, respectively

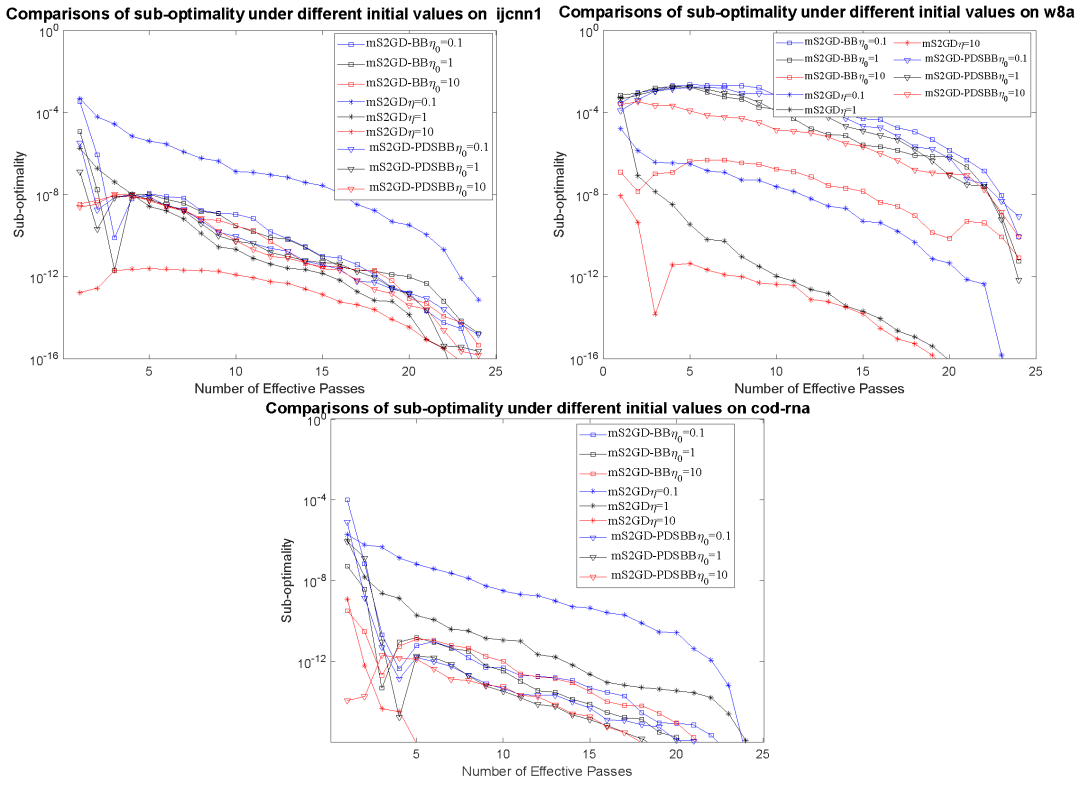

FIGURE 5. The sub-optimality about mS2GD-BB, mS2GD and mS2GD-PDSBB with different initial step size on ijcnn1, w8a, and cod-rna, respectively

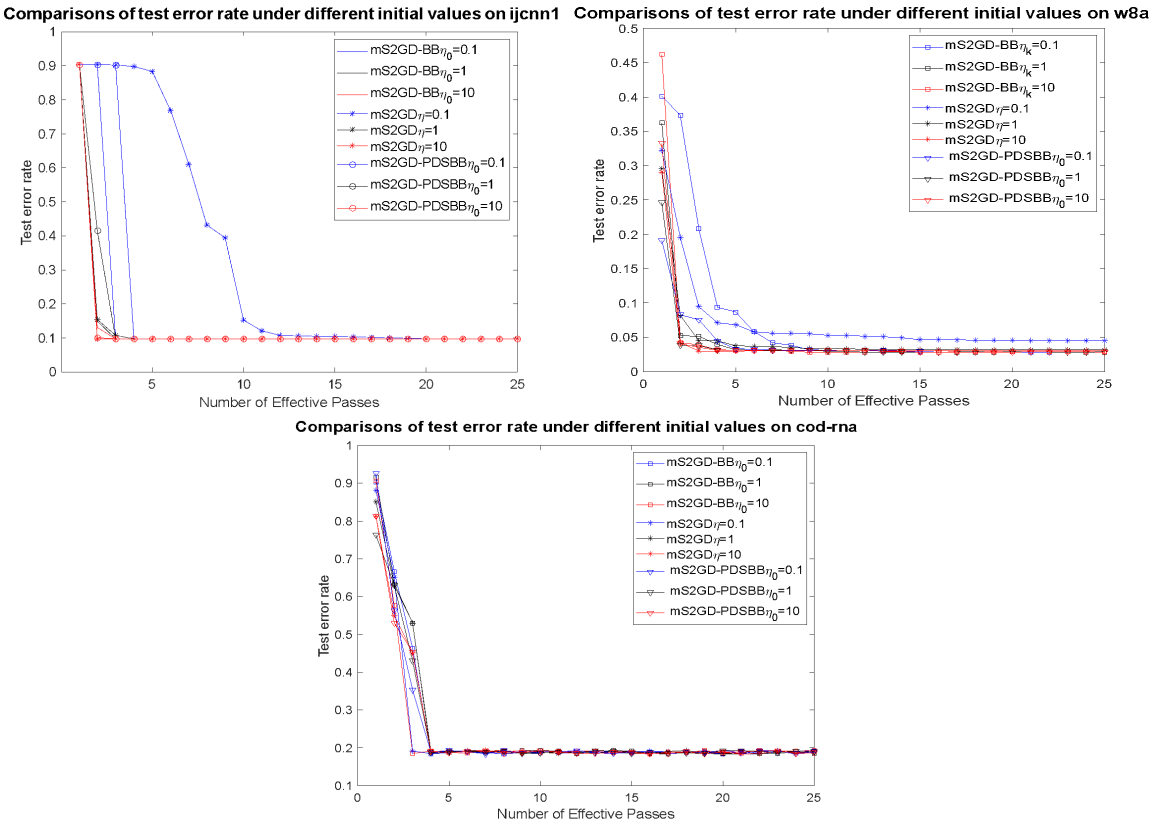

FIGURE 6. The test error rate about mS2GD-BB, mS2GD and mS2GD-PDSBB with different initial step size on ijcnn1, w8a, and cod-rna, respectively

CONCLUSION

In this paper, a new mini-batch algorithm named mS2GD-PDSBB are introduced. A detailed examination of the mS2GD-PDSBB algorithm is provided. The pseudo code for the algorithm is provided to facilitate a better understanding by other researchers. Theoretical analyses are provided to demonstrate that mS2GD-PDSBB achieves linear convergence when applied to non-smooth objective functions. A series of numerical experiments were conducted, offering a detailed analysis concerning various datasets, ijcnn1, w8a, and cod-rna, along with varying initial step sizes (η_0 = 0.1,1, or 10). This paper primarily focuses on aspects like step size, sub-optimality, and error rate to gain deeper insights into the algorithm's performance under different conditions.

Based on the research findings, the conclusion is drawn that the mS2GD-PDSBB algorithm exhibits efficiency across multiple scenarios, showing insensitivity to the initial step size. The numerical results indicate that the step size of the mS2GD-PDSBB algorithm converges to a near-optimal value within a few iterations. Moreover, comparison with other optimization algorithms demonstrates the effectiveness of mS2GD-PDSBB. These findings highlight that mS2GD-PDSBB is both efficient and robust across various scenarios. The proposed approach provides a practical and theoretically sound tool for large-scale optimization problems, contributing a novel adaptive step size strategy to the field. Based on the current research results, future work can focus on the theoretical properties of the proposed mS2GD-PDSBB algorithm in non-convex environments, and explore its application in practical large-scale optimization problems.

ACKNOWLEDGMENTS

This work was supported by Scientific Research Program of Hunan Provincial Education Department in 2024.

REFERENCES

Barzilai, J. & Borwein, J.M. 1988. Two-point step size gradient methods. *IMA Journal of Numerical Analysis* 8(1): 141-148.

Berahas, A.S., Nocedal, J. & Takáč, M. 2016. A multi-batch L-BFGS method for machine learning. *Advances in Neural Information Processing Systems*. pp. 1063-1071.

Chen, G., Li, Y., Zhang, J. & Huang, K. 2022. A survey of the four pillars for small object detection. *Foundations and Trends in Computer Graphics and Vision* 14(1-2): 1-145.

Condat, L. 2023. Proximal splitting algorithms for convex optimization: A tour of recent advances. *SIAM Review* 65(3): 699-763.

Hosny, S., Shouman, M.A. & Ali, A.A. 2023. Survey on compressed sensing over the past two decades. *Array* 19: 100308.

Khan, S., Naseer, M., Hayat, M., Zamir, S.W., Khan, F.S. & Shah, M. 2022. Transformers in vision: A survey. *ACM Computing Surveys* 54(10): 200.

Konečný, J., Liu, J., Richtárik, P. & Takáč, M. 2016. Mini-batch semi-stochastic gradient descent in the proximal setting. *IEEE Journal on Selected Topics in Signal Processing* 10(2): 242-255.

Li, M., Zhang, T., Chen, Y. & Smola, A.J. 2014. Efficient mini-batch training for stochastic optimization. *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 661-670.

Lin, T., Wang, Y., Liu, X. & Qiu, X. 2022. A survey of transformers. *AI Open* 3: 111-132.

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X. & Pietikäinen, M. 2020. Deep learning for generic object detection: A survey. *International Journal of Computer Vision* 128: 261-318.

Ma, K., Zeng, J., Xiong, J., Xu, Q., Cao, X., Liu, W. & Yao, Y. 2018. Stochastic non-convex ordinal embedding with stabilized Barzilai-Borwein step size. *Proceedings of the 32nd AAAI Conference on Artificial Intelligence* 458: 3738-3745.

Mittal, P., Ghosh, A. & Singh, S.K. 2024. A comprehensive survey of deep learning-based lightweight object detection on edge devices. *Artificial Intelligence Review* 57: 242.

Nagahara, M. 2024. A survey on compressed sensing approach to systems and control. *SN Computer Science* 5(5): 442.

Nesterov, Y. 2013. Gradient methods for minimizing composite functions. *Mathematical Programming* 140(1): 125-161.

Nguyen, L.M., Liu, J., Scheinberg, K. & Takáč, M. 2017. SARAH: A novel method for machine learning problems using stochastic recursive gradient. *Proceedings of the 34th International Conference on Machine Learning*. pp. 4009-4022.

Parikh, N. & Boyd, S. 2014. Proximal algorithms. *Foundations and Trends in Optimization* 1(3): 127-239.

Reddi, S.J., Hefny, A., Sra, S., Póczós, B. & Smola, A. 2016. Stochastic variance reduction for nonconvex optimization. *Proceedings of the 33rd International Conference on Machine Learning.* pp. 314-323.

SAS. 2023. *Machine Learning*: What it is and why it matters.

Shao, Y., Wang, Q. & Han, D. 2022. Efficient methods for convex problems with Bregman Barzilai–Borwein step sizes. *Pacific Journal of Optimization* 18(2): 333-348.

Shi, W., Shuib, A. & Alwadood, Z. 2023. Stochastic variance reduced gradient method embedded with positive defined stabilized Barzilai–Borwein. *IAENG International Journal of Applied Mathematics* 53(4): 1682-1687.

Tseng, P. 2000. A modified forward-backward splitting method for maximal monotone mappings. *SIAM Journal on Control and Optimization* 38(2): 431-446.

Wen, L., Cheng, Y., Fang, Y. & Li, X. 2023. A comprehensive survey of oriented object detection in remote sensing images. *Expert Systems with Applications* 224: 119960.

Xiao, L. & Zhang, T. 2014. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization* 24(4): 2057-2075.

Yang, Z. 2024. SARAH-M: A fast stochastic recursive gradient descent algorithm via momentum. *Expert Systems with Applications* 238: 122295.

Yang, Z., Chen, Z. & Wang, C. 2021. Accelerating mini-batch SARAH by step size rules. *Information Sciences* 558: 157-173.

Yang, Z., Wang, C., Zhang, Z. & Li, J. 2019. Mini-batch algorithms with online step size. *Knowledge-Based Systems* 165: 228-240.

Yang, Z., Wang, C., Zhang, Z. & Li, J. 2018. Random Barzilai–Borwein step size for mini-batch algorithms. *Engineering Applications of Artificial Intelligence* 72: 124-135.

*Corresponding author; email: adibah253@uitm.edu.my