Electronic Research Archive

https://www.aimspress.com/journal/era

ERA, 34(6): 3736–3767.
DOI: 10.3934/era.2026169
Received: 28 December 2025
Revised: 27 March 2026
Accepted: 08 April 2026
Published: 06 May 2026

Research article

A biased stochastic three-term conjugate gradient algorithm with bandwidth-based step size for machine learning

Shuang Wang[1], **Gonglin Yuan**[1], **Junyu Lu**[1] **and Yiqian Wei**[2,*]

[1] School of Mathematics & Center for Applied Mathematics of Guangxi & Post-doctoral Research Station of the First-level Discipline in Mathematics, Guangxi University, Nanning 530004, China

[2] Center for Faculty and Teaching Development, Guangxi Vocational University of Agriculture, Nanning 530007, China

* **Correspondence:** Email : weiyiqian_academic@163.com.

Abstract: In the field of machine learning, the solution of large-scale data optimization problems faces numerous challenges. Traditional conjugate gradient (CG) algorithms, though possessing excellent convergence properties, incur high computational costs when dealing with large-scale problems, thereby restricting their scope of application. On the other hand, stochastic gradient descent (SGD) algorithm, while being computationally inexpensive, has its convergence rates limited by the variance of gradient estimates, making it difficult to achieve satisfactory optimization results. To address these limitations, this paper proposes a biased stochastic three-term conjugate gradient algorithm. The proposed algorithm integrates the stochastic recursive gradient algorithm (SARAH) and a bandwidth-based step size strategy. Without incurring additional computational costs, it automatically incorporates upper and lower bounds on the step size, effectively balancing the flexibility and stability of the step size. Through the theoretical analysis presented in this paper, we demonstrate that the algorithm converges to a global optimum and analyze the linear convergence rate of the non-convex (λ-gradient dominated) objective functions. The numerical results of two machine learning models demonstrate the strong competitiveness from the biased stochastic three-term conjugate gradient algorithm with bandwidth-based step size (SCGBW) algorithm.

Keywords: machine learning; stochastic conjugate gradient; SARAH algorithm; bandwidth-based step size; three-term conjugate gradient method; non-convex problems

1. Introduction

Within the realm of machine learning, the empirical risk minimization (ERM) model is one of the core methods used for constructing predictive models. This model seeks for the optimal model parameters by minimizing the empirical risk on the training data, thereby achieving accurate predictions

on unseen data. Specifically, the ERM model can be formulated as the following optimization problem:

$$\min_{\omega\in\Re^d} \phi(\omega) = \frac{1}{N}\sum_{j=1}^{N} \phi_j(\omega) + m(\omega), \tag{1.1}$$

where the objective function $\phi_j(\omega) : \Re^d \to \Re$, $j \in 1, ..., N$ corresponds to the jth loss function. The N is expressed as the number of samples, and $m(\omega)$ is lower semicontinuous (but possibly non-differentiable) function, often referred to as the proximal function. In many practical applications, $m(\omega)$ is commonly used to introduce regularization terms to prevent model overfitting. For example, in ridge regression, $m(\omega)$ is $\frac{\lambda}{2}\|\omega\|_2^2$; in Lasso Regression, $m(\omega)$ is $\lambda\|\omega\|_1$. The full gradient descent algorithm, initially proposed by Cauchy [1], is a classic approach to tackle this issue. The central idea is to compute the gradient over the entire dataset to guide parameter adjustments. Each update is a move against this gradient, a process that systematically lowers the loss function, aligning with the core objective of minimization. The optimization procedure repeats until a local or global optimum is found. The rise of machine learning, marked by its swift development and extensive use, has been facilitated by the massive growth in data and persistent improvements in computational capacity. Consequently, the study and processing of algorithms for large-scale samples have become particularly important. In order to solve the large-scale machine learning problems, stochastic gradient decent (SGD) [2] has become a highly popular method for solving (1.1). Several adaptive optimizers have been developed to advance beyond standard SGD, with the adaptive subgradient method (AdaGrad), the root mean square propagation technique (RMSProp), and the adaptive moment estimation approach (Adam) being prominent examples. A key contribution of AdaGrad [3] is its automation of per-parameter step size configuration, achieved by accumulating the squares of past gradients. This method obviates the extensive manual search for an optimal fixed learning rate, a common challenge when applying the basic SGD algorithm. Expanding upon AdaGrad, RMSProp [4] proposes a learning rate that is dynamically adjusted in accordance with the observed gradient information. Building further on this foundation, Adam, introduced by Kingma and Ba [5], enhances RMSProp by integrating a momentum term. Dozat [6] augmented the Adam algorithm by incorporating Nesterov momentum, thereby markedly enhancing the convergence rate and the overall learning performance of models. The preference for SGD stems from its economical computational cost, which is attained by substituting the true full gradient with a gradient estimator. However, as iterations progress, its convergence rate is capped at $\frac{1}{\epsilon}$, due to the variance introduced by the estimator. To tackle this issue, various algorithms have been explored. The stochastic average gradient (SAG) algorithm [7] computes the average of random gradient values rather than relying exclusively on the most recent gradient value. However, its high storage cost when dealing with large sample sizes limits its practical application. The SAGA algorithm [8] overcomes this by providing a more accurate conditional expectation of the gradient, thereby enhancing theoretical properties. Introduced by Johnson and Zhang, the stochastic variance reduced gradient (SVRG) [9] algorithm incorporates a specialized variance reduction strategy. This technique is notable for its ability to accelerate the convergence of stochastic first-order optimization methods. The SVRG effectively reduces the variance of gradient estimates to zero as the iterations increase. Nguyen et al. [10] proposed a mini-batch variant named SARAH, which demonstrates a linear convergence rate for strongly convex objective functions. Notably, SARAH's inner loop converges, and its gradient estimator is biased, setting it apart from the SVRG. With the development of stochastic optimization methods such as SGD, SVRG, and

SARAH, and to further enhance the efficiency and stability of optimization algorithms in large-scale data scenarios, researchers have introduced the concept of a conjugate gradient algorithm into the stochastic optimization framework, leading to the development of the stochastic conjugate gradient(CG) algorithm. By integrating the efficient convergence properties of conjugate directions with the computational advantages of stochastic sampling, this approach effectively reduces computational complexity while maintaining convergence speed, providing another important tool for solving large-scale machine learning problems.

1.1. Related works

- **Conjugate gradient algorithms.** CG algorithms are highly efficient for solving large-scale optimization problems due to their low memory requirements and strong numerical performance. The foundational work by Hestenes and Stiefel (HS) [11] introduced the CG algorithm for solving linear systems, including both symmetric and asymmetric cases. Later, Fletcher and Reeves (FR) [12] extended this approach to nonlinear optimization, marking a significant advancement in numerical optimization. Further improvements were made by Dai and Yuan (DY) [13], who developed a class of CG algorithms that guarantee global convergence under the standard Wolfe conditions. Around the same time, Polyak [14] and Polak and Ribi$\grave{e}$re [15] proposed an enhanced version of the CG algorithm called the Polak–Ribiére–Polyak (PRP) algorithm, which demonstrated superior performance in numerical experiments. Andrei [16] later introduced a convex combination of the HS and DY algorithms, outperforming both original algorithms in practice. For nonconvex optimization, Raydan [17] combined the Barzilai–Borwein(BB) gradient algorithm with a nonmonotone line search, proving its effectiveness in challenging cases. Meanwhile, Yu et al. [18] modified the spectral CG algorithm proposed by Birgin and Martinez [19], ensuring a sufficient descent property that was independent of the line search conditions. In 2007, the pioneering three-term CG formula was introduced by Zhang et al. [20].
- **Stochastic conjugate gradient algorithms.** The high computational complexity of the CG method in large-scale and complex modeling scenarios has motivated the recent development of numerous stochastic conjugate gradient (SCG) algorithms. Because the traditional CG algorithm has a high computational complexity when dealing with large-scale data and complex models, over the past few years, numerous researchers have introduced SCG algorithms. For instance, Schraudolph and Graepel [21] explored CG algorithms in a stochastic context. They generated Krylov subspaces of lower dimensions within individual minibatches using fast Hessian-gradient techniques. This approach enabled their algorithms to converge more rapidly than conventional SGD methods. Jiang and Wilford [22] put forward a novel SCG algorithm. It bypasses the necessity to compute and store the covariance matrix for least squares solutions. Alternatively, it used an inner product derived from random sampling to ascertain the step size, eschewing the meticulous selection of a fixed step. Byrd et al. [23] developed a Newton-CG algorithm. This approach utilizes distinct subsamples to approximate the function and gradient values. Moreover, it dynamically adjusts the sample size in response to the variance of the gradient estimates, consequently setting a complexity bound for the proposed method. Inspired by both the CG and SVRG algorithms, Jin and Zhang [24] introduced a SCG algorithm with variance reduction (CGVR). In this algorithm, the CG is used in lieu of the general gradient, which significantly enhances the convergence rate. More recently, Alnowibet et al. [25] proposed a new CG algorithm, which showes great promise and

competitiveness in searching for local optima. Yuan et al. [26] provided a novel perspective on SCG algorithms through the lens of stochastic differential equations, offering a rigorous continuous-time analysis framework. Ouyang et al. [27] proposed a new search direction for the three-term conjugate gradient, combined it with SAGA, and obtained the same convergence rate as SCGA (an algorithm obtained by combining CG and SAGA) under weaker conditional assumptions.

- **Adaptive step size strategies.** The choice of step size is critical for the performance of any first-order optimization algorithm. While constant or diminishing step sizes are common, they often require tedious manual tuning to achieve optimal performance. Adaptive strategies that automatically compute the step size are therefore highly desirable. The BB [28] method is a popular technique that leverages second-order information from the most recent iteration at low cost. A recent innovative approach is the bandwidth-based step size proposed by Wang and Yuan [29]. This strategy confines the step size within a banded region, ensuring automatic upper and lower bounds that enhance stability and flexibility without manual intervention. Our work incorporates this bandwidth-based strategy to dynamically adjust the step size in our algorithm, eliminating the need for expensive line searches or sensitive hyperparameter tuning.
- **Synthesis and our algorithm.** Our proposed a biased stochastic three-term conjugate gradient algorithm with bandwidth-based step size (SCGBW) is situated at the intersection of several active research streams: Stochastic optimization with variance reduction, conjugate gradient methods, and adaptive step size strategies. Our algorithm distinguishes itself through the following key components: (i) The SARAH estimator for effective variance reduction, (ii) a modern three-term conjugate gradient framework for efficient computation of the search direction, and (iii) a bandwidth-based step size for stable and adaptive learning. Furthermore, we introduce a novel modification by replacing the standard Y_j with Y_j^m, which incorporates both gradient and function value information, inspired by works on quasi-Newton algorithms [30, 31].

1.2. Our contributions

By combining SARAH and the bandwidth-based step size with the improved stochastic three-term conjugate gradient algorithm, a biased stochastic three-term conjugate gradient algorithm (SCGBW) is proposed. The main contributions of this study are as follows.

1) We first integrate the three-term CG with the SARAH algorithm. Contrary to conventional stochastic gradient techniques that determine the step size through meticulous selection of a constant or via line search methods, we use a bandwidth-based step size, which incorporates an adaptive step size. This not only utilizes second-order information to dynamically adjust the step size but also ensures that the step size to automatically remains within a specified bounds.

2) The conventional Y_j only contains gradient information. We replace the conventional Y_j with a modified Y_j^m, which not only includes gradient information but also incorporates function value information.

3) Under the assumptions of gradient Lipschitz continuity and λ-gradient dominance, through rigorous theoretical analysis, we have established the algorithm's convergence to global optimization and further demonstrated its linear convergence rate when applied to a class of non-convex functions.

4) We evaluate the proposed method using non-convex support vector machine (SVM) problems and the non-convex empirical risk minimization (ERM) problem, with the experimental results showing the

following (i) The selection of the initial step size has a relatively minor impact on the experimental outcomes of the algorithm. (ii) When the initial step size is small, the algorithm we designed exhibits a faster rate of descent and reaches a stable point more quickly compared with other classical algorithms. In summary, the algorithm we designed is highly competitive.

1.3. Work plan

In the Section 2 of this paper, we will present the core motivation behind this research and provide a comprehensive exposition of the proposed algorithm. This includes a detailed breakdown of its key components: The SARAH estimator for variance reduction, the three-term CG framework, and the bandwidth-based step size strategy. Section 3 is dedicated to a rigorous convergence analysis, with particular emphasis on establishing the linear convergence rate of the algorithm under suitable conditions. Theoretical guarantees are provided in the form of detailed proofs and discussions on the influence of key parameters. Finally, the performance of the derived algorithm is evaluated through extensive numerical experiments against several classic optimization algorithms in Section 4, applied to two prevalent machine learning models. A detailed analysis and interpretation of the experimental results are provided to demonstrate the efficiency, robustness, and superiority of the proposed algorithm.

2. Motivation

2.1. The gradient estimator of the SARAH algorithm

Nguyen et al. [10] proposed the SARAH algorithm, tailored for tackling challenges in machine learning. Recent extensions, such as the work by Mo et al. [32] , demonstrate that integrating inertial extrapolation with SARAH can further enhance the stability of convergence in non-convex scenarios. The specific implementation of SARAH is detailed in Algorithm 1.

Algorithm 1 SARAH

Given the step size η_0, the initial loop size m and the initial point ω_0
for $t = 1, 2, \ldots, \infty$ **do**
  $\widetilde{\omega}_0 = x_{t-1}$
  $\zeta_0 = \nabla\phi(\widetilde{\omega}_0)$
  $\widetilde{\omega}_1 = \widetilde{\omega}_0 - \eta_0\zeta_0$
  for $j = 1, 2, \ldots, m-1$ **do**
    Update the step size, randomly select a mini-batch $H_1 \in \{1, 2, \ldots, N\}$ of size h_1, calculate
    $\zeta_j = \nabla\phi_{H_1}(\widetilde{\omega}_j) - \nabla\phi_{H_1}(\widetilde{\omega}_{j-1}) + \zeta_{j-1}$
    $\widetilde{\omega}_{j+1} = \widetilde{\omega}_j - \eta'_j\zeta_j$
  end for
  $\omega_t = \widetilde{\omega}_n$, n is selected randomly from $0, 1, \ldots, m$.
end for

SARAH represents a modification of the SVRG algorithm. SVRG reduces variance by progressively lowering the upper limit of variance in each iteration. In contrast, SARAH introduces an innovative approach, characterized by a critical step that leverages a recursive gradient estimation method within its inner loop. The key update formulas of the SARAH algorithm are as follows:

(i) Its recursive update formula for the stochastic gradient estimate is

$$\zeta_j = \nabla\phi_{H_1}(\widetilde{\omega}_j) - \nabla\phi_{H_1}(\widetilde{\omega}_{j-1}) + \zeta_{j-1}.$$

(ii) The iteration update formula is

$$\widetilde{\omega}_{j+1} = \widetilde{\omega}_j - \eta_j\zeta_j,$$

where $\nabla\phi_{H_1}(\widetilde{\omega}_j) = \frac{1}{h_1}\sum_{i\in H_1}\nabla\phi_i(\widetilde{\omega}_j)$, and $H_1 \in \{1,\ldots,N\}$ with $h_1 = |H_1|$. The parameters of the SARAH algorithm include the learning rate $\eta_0 > 0$ and the inner loop size m. It is initialized with $\widetilde{\omega}_0$, and then iterations are performed. In each outer loop, first set $\widetilde{\omega}_0 = x_{t-1}$, calculate the full gradient $\zeta_0 = \frac{1}{N}\sum_{j=1}^{N}\nabla\phi_j(\omega_0)$, and then perform one gradient descent $\widetilde{\omega}_1 = \tilde{\omega}_0 - \eta_0'\zeta_0$. Next, in the inner loop, randomly and uniformly sample H_1 from $1, 2, ..., N$, and update ζ_j and $\widetilde{\omega}_{j+1}$ according to the recursive formula above. Finally, randomly select n from $\{0, 1, \ldots, m\}$ and set $\omega_t = \widetilde{\omega}_n$.The subsample H_1 selection method of SARAH can be seen in [33–35].

Compared with this, the update formula for SVRG is

$$\zeta_j = \nabla\phi_{H_1}(\widetilde{\omega}_j) - (\nabla\phi_{H_1}(\bar{\omega}) - \zeta_0),$$

where $\zeta_0 = \nabla\phi(\bar{\omega})$, which represent a full gradient of $\bar{\omega}$ and $\bar{\omega}$ is viewed as a snapshot point. It should be noted that ζ_j in SVRG is an unbiased estimator of the gradient, while that in SARAH is not. Specifically:

$$\mathbf{E}[\zeta_j] \neq \nabla\phi(\widetilde{\omega}_j).$$

2.2. Stochastic three-term CG algorithm

Considering the following classical unconstrained optimization problem:

$$\min\{g(\omega) \mid \omega \in \mathbb{R}^d\},$$

where $g : \Re^d \to \Re$. The CG algorithm demonstrates superior capabilities in search direction generation and step size selection when solving large-scale linear systems and non-convex problems. In recent years, the three-term CG method has further extended traditional CG approaches by introducing an additional correction term, which enhances direction diversity and convergence robustness. The first three-term CG formula is presented by Zhang et al. [20] for continuous optimization problems (1.1).

$$d_j = \begin{cases} -\nabla g(\omega_j) + \frac{\nabla g(\omega_j)^T y_j d_{j-1} - \nabla g(\omega_{j-1})^T d_{j-1} y_j}{\|\nabla g(\omega_{j-1})\|^2}, & \text{if } j \geq 1 \\ -\nabla g(\omega_j), & \text{if } j = 0. \end{cases}$$

It is equivalent to

$$d_j = -\nabla g(\omega_j) + \beta_j d_{j-1} - \theta_j y_j,$$

where $y_j = \nabla g(\omega_j) - \nabla g(\omega_{j-1})$ and $\beta_j = \frac{\nabla g(\omega_j)^T y_j}{\|\nabla g(\omega_{j-1})\|^2}, \theta_j = \frac{\nabla g(\omega_j)^T d_{j-1}}{\|\nabla g(\omega_{j-1})\|^2}$.

Existing studies have demonstrated the favorable numerical properties of the three-term CG method. Kim et al. [36] applied this approach to the training of artificial neural networks, where it exhibited competitive performance in comprehensive comparisons with optimizers such as SGD, Adam, AMSGrad [37], and AdaBelief Zhuang et al. [38], although its theoretical foundations remain less explored.

In the context of variance-reduced optimization, several hybrid methods have been developed: Kou and Yang [39] integrated the CG method into the SAGA framework, resulting in the SCGA algorithm. Yang [40] combined mini-batch SARAH [10] with the FR conjugate gradient method to propose CG-SARAH-SO, and Huang et al. [41] introduced BSCG by incorporating a modified PRP method with SARAH. With an increase in the data volume, based on the good properties of three-term conjugate gradient, Ouyang et al. [27] proposed a new direction inspired by the three-term and integrated it with the SAGA algorithm, resulting in a stochastic three-term CG algorithm. The specific formulation is shown below:

$$D_j^* = \begin{cases} -G_j + B_j D_{j-1}^* - O_j Y_j^*, & \text{if } j \geq 1 \\ -G_j, & \text{if } j = 0. \end{cases} \tag{2.1}$$

where D_j^* represents the descending direction of the sample iteration, G_j is the gradient estimated by SAGA, $Y_j^* = G_j - G_{j-1}$, and

$$B_j^* = \frac{G_j^T Y_j^*}{\mu_1 ||Y_j^*|| ||D_{j-1}^*|| + \mu_2 ||Y_j^*|| ||G_j|| + ||G_{j-1}||^2},$$

$$O_j^* = \frac{G_j^T D_{j-1}^*}{\mu_1 ||Y_j^*|| ||D_{j-1}^*|| + \mu_2 ||Y_j^*|| ||G_j|| + ||G_{j-1}||^2}.$$

The denominator in the decent direction of B_j^* and O_j^* is the crucial component. It not only ensures that each step is a valid descent direction but also maintains the trust region property, thereby ensuring the stability of the algorithm.

Compared with the SAGA algorithm, the SARAH algorithm uses a recursive gradient estimation mechanism that achieves smoother variance reduction through the accumulation of historical gradient information, leading to more stable convergence, while requiring only the storage of current parameters and gradients, thereby significantly reducing memory requirements. Inspired by this, we attempt to integrate the new three-term CG search direction (2.1) proposed by Ouyang with the SARAH stochastic recursive gradient estimator, yielding a novel stochastic three-term CG search direction:

$$D_j = \begin{cases} -\zeta_j + B_j D_{j-1} - O_j Y_j, & \text{if } j \geq 1 \\ -\zeta_j, & \text{if } j = 0. \end{cases}$$

where ζ_j is the gradient value estimated by the aforementioned SARAH algorithm and D_j represents the descending direction of the sample iteration, $Y_j = \zeta_j - \zeta_{j-1}$ and

$$B_j = \frac{G_j^T Y_j}{\mu_1 ||Y_j|| ||D_{j-1}|| + \mu_2 ||Y_j|| ||\zeta_j|| + ||\zeta_{j-1}||^2},$$

$$O_j = \frac{G_j^T D_{j-1}}{\mu_1 ||Y_j|| ||D_{j-1}|| + \mu_2 ||Y_j|| ||\zeta_j|| + ||\zeta_{j-1}||^2}.$$

2.3. The bandwidth-based step size

It is widely acknowledged that choosing a suitable step size is vital for the optimal performance of the aforementioned algorithm. When a constant step size $\eta_j = \eta$ is used for SGD, it demonstrates

sublinear convergence to the vicinity of the solution for strongly convex objectives and to a stationary point for non-convex objectives [42, 43]. To accelerate the convergence rate of SGD, diminishing step sizes that meet the following convergence criteria are frequently adopted:

$$\sum_{j=1}^{\infty} \eta_j = +\infty, \quad \sum_{j=1}^{\infty} \eta_j^2 < +\infty.$$

For instance, consider the step size $\eta_j = \frac{\eta_0}{j}$, where η_0 denotes the initial step size. When dealing with a strongly convex function $\phi(\omega)$, SGD converges to the solution at a sublinear rate [44]. When it comes to non-convex functions $\phi(\omega)$, SGD with a step size of $\eta_j = \frac{\eta_0}{\sqrt{j}}$ exhibits sublinear convergence to a stationary point [45]. By contrast, SVRG utilizing a constant step size $\eta_j \equiv \eta$ achieves convergence to the solution for both strongly convex and non-convex objectives. Furthermore, when the objective function is strongly convex, SVRG converges at a linear rate [9, 46]. Nevertheless, manually tuning the aforementioned constant step size to reach optimal performance is a time-consuming process.Therefore, techniques for automatically determining the step size during the algorithm's execution have been devised. One popular selection is the BB step size [28]. The BB methods, put forward by Barzilai and Borwein [28], target the determination of the step size with diminished computational and storage costs. It functions as an adaptive step size selection technique used in optimization algorithms. Different from other line search methods that may require a lot of historical data, the BB method uniquely updates the step size solely on the basis of information from two consecutive points. This streamlined approach simplifies the computational process and enhances the convergence rate, especially when tackling intricate non-convex functions. However, in some iterations, the BB step size may cause the algorithm to jump out of the neighborhood of the optimal solution and even diverge. We usually need to introduce upper and lower bounds on the step size to limit the range of the BB step size. Recently, Wang and Yuan [29] proposed a novel concept known as the bandwidth-based step size. This approach allows the step size to vary within a specific banded region and automatically remain within specified bounds, i.e.,

$$b\delta(j) \leq \eta_j \leq B\delta(j) \quad j \geq 1,$$

where $b \leq B$ are positive constants, which is the bound of the step size. The boundary function $\delta(j)$ satisfies the aforementioned convergence conditions [47]:

$$\sum_{t=1}^{\infty} \delta(j) = +\infty, \quad \lim_{t \to +\infty} \delta(j) = 0, \quad \frac{d\delta(j)}{dj} \leq 0.$$

If we let $\delta(j) = 1/j$, it is obviously satisfied by the convergence conditions. Mini-batch setting should be considered when utilizing the bandwidth-based step size. Similar to the approach in [41] dividing the sample size h_2 reduces the effect of choosing subsamples randomly, so for stochastic optimization problems, we have

$$\eta_j' = \begin{cases} \frac{b}{j}, & \text{if } \eta_j^{BB^*} \leq \frac{b}{j}; \\ \eta_j^{BB^*}, & \text{if } \frac{b}{j} < \eta_j^{BB^*} < \frac{B}{j}; \\ \frac{B}{j}, & \text{if } \eta_j^{BB^*} \geq \frac{B}{j}. \end{cases} \tag{2.2}$$

In (2.2), $\eta_j^{BB^*} = \frac{1}{h_2} \frac{\|s_j\|^2}{s_j^T Y_j}$, and $s_j = \omega_j - \omega_{j-1}$.

2.4. A stochastic three-term CG algorithm with parameter correction and bandwidth-based step size

Huang et al. [41] applied SARAH and the modified BB step size to CG methods and obtained competitive results for noncovex problems.

In the general three-term CG methods, the formula for y_j only incorporates gradient information. Recently, new formulations for y_j have been proposed. For example, Wei and Yuan [30] introduced a new Broyden-Fletcher-Goldfarb-Shanno (BFGS) method in which

$$y_j^* = y_j + \frac{\rho_j}{\|s_j\|^2} s_j,$$

where $\rho_j = 2(g(\omega_{j-1}) - g(\omega_j)) + (\nabla g(\omega_j) + \nabla g(\omega_{j-1}))^T s_{j-1}$. It can be seen that y_j^* incorporates both gradient information and function value information. Yuan and Wei [31] improved upon the aforementioned BFGS formula and proposed a modified BFGS formula, which is

$$y_j^m = y_j + \frac{\max\{\rho_j, 0\}}{\|s_j\|^2} s_j. \tag{2.3}$$

We try to apply (2.3) to the stochastic optimization problem:

$$Y_j^m = Y_j + \frac{\max\{p_j, 0\}}{\|s_j\|^2} s_j, \tag{2.4}$$

where $p_j = 2(\phi(\omega_j) - \phi(\omega_{j-1})) + (\zeta_j + \zeta_{j-1})^T s_{j-1}$. We draw inspiration from their idea. To enable the step size to vary automatically within a fixed region, we have adopted the bandwidth-based step size. We aim to investigate whether the combination of a bandwidth-based step size and a three-term CG algorithm with better performance, coupled with the SARAH algorithm and extending the y_j^m from the traditional unconstrained optimization to Y_j^m in stochastic optimization, will yield superior outcomes. In this study, we propose a new stochastic three-term CG algorithm that is primarily based on an improved search direction of the three-term CG, incorporating the SARAH algorithm and bandwidth-based step size, in conjunction with the modified Y_j^m. The current algorithm framework is as follows, which is called SCGBW.

Algorithm 2 SCGBW

Given the constants $\eta, m, b, B, h_1, h_2, \mu_1, \mu_2$, and ω_0
for $t = 1, 2, \ldots, \infty$ **do**
  $\tilde{\omega}_0 = \omega_{t-1}$
  $\eta_0' = \eta$
  $\zeta_0 = \nabla\phi(\tilde{\omega}_0)$
  $v_0 = \zeta_0$
  $D_0 = -v_0$
  for $j = 0, 1, 2, \ldots, m$ **do**
    $\tilde{\omega}_{j+1} = \tilde{\omega}_j + \eta_j' D_j$
    Select a mini-batch $H_1 \in \{1, 2, \ldots, N\}$ of size h_1 uniformly at random then calculate
    $\zeta_j = \nabla\phi_{H_1}(\tilde{\omega}_j) - \nabla\phi_{H_1}(\tilde{\omega}_{j-1}) + \zeta_{j-1}$
    calculate Y_j^m by (2.4)
    $B_j = \frac{\zeta_j^T Y_j^m}{\mu_1 \|Y_j^m\| \|D_{j-1}\| + \mu_2 \|Y_j^m\| \|\zeta_j\| + \|\zeta_{j-1}\|^2}$
    $O_j = \frac{\zeta_j^T D_{j-1}}{\mu_1 \|Y_j^m\| \|D_{j-1}\| + \mu_2 \|Y_j^m\| \|\zeta_j\| + \|\zeta_{j-1}\|^2}$
    Determine $D_j = -\zeta_j + B_j^T D_{j-1} - O_j^T Y_j^m$
    Select a mini-batch $H_2 \in \{1, 2, \ldots, N\}$ of size h_2 uniformly at random then calculate
    $$\eta_j' = \begin{cases} \frac{b}{j}, & \text{if } \eta_j^{BB^*} \le \frac{b}{j}; \\ \eta_j^{BB^*}, & \text{if } \frac{b}{j} < \eta_j^{BB^*} < \frac{B}{j}; \\ \frac{B}{j}, & \text{if } \eta_j^{BB^*} \ge \frac{B}{j}. \end{cases}$$
  end for
  $u_t = \zeta_m$
  $\omega_t = \tilde{\omega}_m$
end for

3. Global convergence analysis

3.1. Assumption

Assumption I: The loss functions are twice continuous differentiable with a Lipschitz continuous gradient such that:

$$\|\nabla\phi_j(\omega_1) - \nabla\phi_j(\omega_2)\| \le L\|\omega_1 - \omega_2\|, \quad \omega_1, \omega_1 \in \mathbf{R}^d, \forall j \in 1, ..., N.$$

According to Assumption I, the following significant inequalities can be readily derived:

$$\phi(\omega_1) \le \phi(\omega_2) + \langle\nabla\phi(\omega_2), \omega_1 - \omega_2\rangle + \frac{L}{2}\|\omega_1 - \omega_2\|^2.$$

Assumption II: (λ-gradient dominated) ϕ is λ-gradient dominated, i.e., it means that $\forall\omega \in \mathbf{R}^d$, and there is a constant $\lambda > 0$ satisfying:

$$\phi(\omega) - \phi(\omega_*) \le \lambda\|\nabla\phi(\omega)\|^2,$$

where ω_* is a global minimum point of ϕ. On the basis of Assumption II, it can be easily inferred that every stationary point of the function ϕ is a global minimizer. Importantly, the gradient-dominated function, a class of non-convex functions introduced by Polyak [48], has been widely utilized by Nesterov and Polyak [49] as well as Reddi et al. [46].

3.2. Convergence analysis

Lemma 3.1. *Assuming that Assumption I is valid, we see that the step size is bounded.*

Proof. The step length obviously satisfies the inequality $b \leq \eta'_j \leq B$. Among them, B and b are fixed positive numbers. We set $\eta'_{\max} = B$ and $\eta'_{\min} = b$. □

Lemma 3.2. *Assuming that Assumption I holds and considering that ζ_j is defined by SARAH, $\forall\, j \geq 1$, we obtain*

$$\mathbb{E}[\|\nabla\phi(\tilde{\omega}_j) - \zeta_j\|^2] \leq \sum_{k=1}^{j} \mathbb{E}[\|\zeta_k - \zeta_{k-1}\|^2] + \sum_{k=1}^{j} \mathbb{E}[\|\nabla\phi(\tilde{\omega}_k) - \nabla\phi(\tilde{\omega}_{k-1})\|^2].$$

Proof. Let $\mathcal{F}_k = \sigma(\tilde{\omega}_0, i_1, i_2, \ldots, i_{k-1})$ be the σ-algebra generated by $w_0, i_1, i_2, \ldots, i_{k-1}$; $\mathcal{F}_0 = \mathcal{F}_1 = \sigma(\tilde{\omega}_0)$. Note that $\mathcal{F}_k$ also contains all the information of $\tilde{\omega}_0, \ldots, \tilde{\omega}_k$ as well as $\zeta_0, \ldots, \zeta_{k-1}$. For $j \geq 1$, we have

$$\begin{aligned}
\mathbb{E}[\|\nabla\phi(\tilde{\omega}_k) - \zeta_k\|^2|\mathcal{F}_k] &= \mathbb{E}[\|[\nabla\phi(\tilde{\omega}_{k-1}) - \zeta_{k-1}] + [\nabla\phi(\tilde{\omega}_k) - \nabla\phi(\tilde{\omega}_{k-1})] - [\zeta_k - \zeta_{k-1}]\|^2|\mathcal{F}_k] \\
&= \|\nabla\phi(\zeta_{k-1}) - \zeta_{k-1}\|^2 + \|\nabla\phi(\tilde{\omega}_k) - \nabla\phi(\tilde{\omega}_{k-1})\|^2 + \mathbb{E}[\|\zeta_k - \zeta_{k-1}\|^2|\mathcal{F}_k] \\
&\quad + 2(\nabla\phi(\tilde{\omega}_{k-1}) - \zeta_{k-1})^T(\nabla\phi(\tilde{\omega}_k) - \nabla\phi(\tilde{\omega}_{k-1})) \\
&\quad - 2(\nabla\phi(\tilde{\omega}_{k-1}) - \zeta_{k-1})^T\mathbb{E}[\zeta_k - \zeta_{k-1}|\mathcal{F}_k] \\
&\quad - 2(\nabla\phi(\tilde{\omega}_k) - \nabla\phi(\tilde{\omega}_{k-1}))^T\mathbb{E}[\zeta_k - \zeta_{k-1}|\mathcal{F}_k] \\
&= \|\nabla\phi(\tilde{\omega}_{k-1}) - \phi_{k-1}\|^2 - \|\nabla\phi(\tilde{\omega}_k) - \nabla\phi(\tilde{\omega}_{k-1})\|^2 + \mathbb{E}[\|\zeta_k - \zeta_{k-1}\|^2|\mathcal{F}_k],
\end{aligned}$$

where the last equality follows from

$$\begin{aligned}
\mathbb{E}[\zeta_k - \zeta_{k-1}|\mathcal{F}_k] &= \mathbb{E}\Big[\frac{1}{h_1}\sum_{i\in H_1}[\nabla\phi_i(\tilde{\omega}_k) - \nabla\phi_i(\tilde{\omega}_{k-1})]\Big|\mathcal{F}_k\Big] \\
&= \frac{1}{h_1}\cdot\frac{h_1}{N}\sum_{i=1}^{N}[\nabla\phi_i(\tilde{\omega}_k) - \nabla\phi_i(\tilde{\omega}_{k-1})] = \nabla\phi(\tilde{\omega}_k) - \nabla\phi(\tilde{\omega}_{k-1}).
\end{aligned}$$

By taking expectation for the equation above, we have

$$\mathbb{E}[\|\nabla\phi(\tilde{\omega}_k) - \zeta_k\|^2] = \mathbb{E}[\|\nabla\phi(\tilde{\omega}_{k-1}) - \zeta_{k-1}\|^2] - \mathbb{E}[\|\nabla\phi(\tilde{\omega}_k) - \nabla\phi(\tilde{\omega}_{k-1})\|^2] + \mathbb{E}[\|\zeta_k - \zeta_{k-1}\|^2].$$

Note that

$$\|\nabla\phi(\tilde{\omega}_0) - \zeta_0\|^2 = 0.$$

By summing over $k = 1, \ldots, j$ ($j \geq 1$), we have

$$\mathbb{E}[\|\nabla\phi(\tilde{\omega}_j) - \zeta_j\|^2] = \sum_{k=1}^{j} \mathbb{E}[\|\zeta_k - \zeta_{k-1}\|^2] - \sum_{k=1}^{j} \mathbb{E}[\|\nabla\phi(\tilde{\omega}_k) - \nabla\phi(\tilde{\omega}_{k-1})\|^2].$$

□

Lemma 3.3. *Assuming that Assumption II holds, and combining Lemma 3.1, $\forall j \geq 1$, we have*

$$\mathbb{E}[\|\nabla\phi(\tilde{\omega}_j) - \zeta_j\|^2] \leq \frac{\eta_{\max}^2 L^2}{h_1}\left(\frac{N-b_1}{N-1}\right)\sum_{k=1}^{j}\mathbb{E}[\|\zeta_{k-1}\|^2].$$

Proof. Let

$$g_j = \nabla\phi_j(\tilde{\omega}_k) - \nabla\phi_j(\tilde{\omega}_{k-1}).$$

We have

$$\begin{aligned}
&\mathbb{E}[\|\zeta_k - \zeta_{k-1}\|^2 \mid \mathcal{F}_k] - \|\nabla\phi(\tilde{\omega}_k) - \nabla\phi(\tilde{\omega}_{k-1})\|^2 \\
&= \mathbb{E}[\|\frac{1}{h_1}\sum_{i\in H_1}[\nabla\phi_i(\tilde{\omega}_k) - \nabla\phi_i(\tilde{\omega}_{k-1})]\|^2|\mathcal{F}_j] - \|\frac{1}{N}\sum_{i=1}^{N}[\nabla\phi_i(\tilde{\omega}_k) - \nabla\phi_i(\tilde{\omega}_{k-1})]\|^2 \\
&= \mathbb{E}[\|\frac{1}{h_1}\sum_{i\in H_1}\xi_i\|^2|\mathcal{F}_j] - \|\frac{1}{N}\sum_{i=1}^{N}\xi_i\|^2 \\
&= \frac{1}{h_1^2}\mathbb{E}[\sum_{i\in H_1}\sum_{t\in H_1} g_i^T g_t|\mathcal{F}_j] - \frac{1}{N^2}\sum_{i=1}^{N}\sum_{t=1}^{N} g_i^T g_t \\
&= \frac{1}{h_1^2}\mathbb{E}[\sum_{i\neq t\in H_1} g_i^T g_t + \sum_{i\in H_1} g_i^T g_i|\mathcal{F}_k] - \frac{1}{N^2}\sum_{i=1}^{N}\sum_{k=1}^{N} g_i^T g_t \\
&= \frac{1}{h_1^2}[\frac{h_1}{N}\frac{(h_1-1)}{(N-1)}\sum_{i\neq t} g_i^T g_t + \frac{h_1}{N}\sum_{i=1}^{N} g_i^T g_i] - \frac{1}{N^2}\sum_{i=1}^{N}\sum_{t=1}^{N} g_i^T g_t \\
&= \frac{1}{h_1^2}[\frac{h_1}{N}\frac{(h_1-1)}{(N-1)}\sum_{i=1}^{N}\sum_{t=1}^{N} g_i^T g_t + \left(\frac{h_1}{N} - \frac{h_1}{N}\frac{(h_1-1)}{(N-1)}\right)\sum_{i=1}^{N} g_i^T g_i] - \frac{1}{N^2}\sum_{i=1}^{N}\sum_{t=1}^{N} g_i^T g_t \\
&= \frac{1}{h_1 N}[\left(\frac{(h_1-1)}{(N-1)} - \frac{h_1}{N}\right)\sum_{i=1}^{N}\sum_{t=1}^{N} g_i^T g_t + \frac{(N-h_1)}{(N-1)}\sum_{i=1}^{N} g_i^T g_i] \\
&= \frac{1}{h_1 N}(\frac{N-h_1}{h_1-1})[-\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{N} g_i^T g_t + \sum_{i=1}^{N} g_i^T g_i] \\
&= \frac{1}{h_1 N}(\frac{N-h_1}{N-1})[-N\|\frac{1}{N}\sum_{i=1}^{N} g_i\|^2 + \sum_{i=1}^{N}\|g_i\|^2] \\
&\leq \frac{1}{h_1}(\frac{N-h_1}{N-1})\frac{1}{N}\sum_{i=1}^{N}\|g_i\|^2 \\
&= \frac{1}{h_1}(\frac{N-h_1}{N-1})\frac{1}{N}\sum_{i=1}^{N}\|\nabla\phi_i(\tilde{\omega}_k) - \nabla\phi_i(\tilde{\omega}_{k-1})\|^2 \\
&\leq \frac{1}{h_1}(\frac{N-h_1}{N-1})L^2\eta_{max}^2\frac{1}{N}\sum_{i=1}^{N}\|\zeta_{k-1}\|^2 \\
&= \frac{1}{h_1}(\frac{N-h_1}{N-1})L^2\eta_{max}^2\|\zeta_{k-1}\|^2.
\end{aligned}$$

Hence, by taking the expectation, we have

$$\mathbb{E}[\|\zeta_k - \zeta_{k-1}\|^2] - \mathbb{E}[\|\nabla\phi(\tilde{\omega}_k) - \nabla\phi(\tilde{\omega}_{k-1})\|^2] \leq \frac{1}{h_1}(\frac{N-h_1}{N-1})L^2\eta_{max}^2\mathbb{E}[\|\zeta_{k-1}\|^2].$$

By Lemma 3.2, for $j \geq 1$,

$$\begin{aligned}\mathbb{E}[\|\nabla\phi((\tilde{\omega}_j) - \zeta_j\|^2] &= \sum_{k=1}^{j}\mathbb{E}[\|\zeta_k - \zeta_{k-1}\|^2] - \sum_{k=1}^{j}\mathbb{E}[\|\nabla\phi((\tilde{\omega}_k) - \nabla\phi((\tilde{\omega}_{k-1})\|^2] \\ &\leq \frac{1}{h_1}(\frac{N-h_1}{N-1})L^2\eta_{max}^2\sum_{k=1}^{j}\mathbb{E}[\|\zeta_{k-1}\|^2].\end{aligned}$$

This completes the proof. □

Lemma 3.4. *We can derive the following property according to the expression of D_j:*

$$G_j^T D_j = -\|G_j\|^2.$$

Proof. When $j = 1$, the above mentioned property obviously holds. For $j > 1$, we have

$$\begin{aligned}\zeta_j^T D_j &= \zeta_j^T\left[-\zeta_j + \frac{(Y_j^m)^T\zeta_j D_{j-1} - D_{j-1}^T\zeta_j(Y_j^m)^T}{\mu_1\|(Y_j^m)^T\|\|D_{j-1}\| + \mu_2\|(Y_j^m)^T\|\|\zeta_j\| + \|\zeta_{j-1}\|^2}\right] \\ &= -\|\zeta_j\|^2 + \frac{(Y_j^m)^T\zeta_j D_{j-1}^T\zeta_j - D_{j-1}^T\zeta_j(Y_j^m)^T\zeta_j}{\mu_1\|(Y_j^m)^T\|\|D_{j-1}\| + \mu_2\|(Y_j^m)^T\|\|\zeta_j\| + \|\zeta_{j-1}\|^2} \\ &= -\|\zeta_j\|^2\end{aligned}$$

From this, it can be concluded that D_j is a descent direction. □

Lemma 3.5. *If we assume that Assumption I holds, we have*

$$\sum_{j=0}^{m}\mathbb{E}[\|\nabla\phi(\tilde{\omega}_j)\|^2] \leq (\frac{2L\eta_{\max}^2}{\eta_{\min}} - 1)\sum_{j=0}^{m}\mathbb{E}[\|\zeta_j\|^2] + \frac{2\eta_{\max}}{\eta_{\min}}\sum_{j=0}^{m}\mathbb{E}[\|\nabla\phi(\tilde{\omega}_j) - \zeta_j\|^2] + \frac{2}{\eta_{\min}}\mathbb{E}[\|\phi(\tilde{\omega}_0) - \phi(\tilde{\omega}_*)\|^2].$$

Proof. Before proving this lemma, let us first introduce two equations. These two equations will be used in the subsequent proof process.

$$\langle\tilde{\omega}_1, \tilde{\omega}_2\rangle = \frac{1}{4}(\|\tilde{\omega}_1 + \tilde{\omega}_2\|^2 - \|\tilde{\omega}_1 - \tilde{\omega}_2\|^2),$$

$$\langle\tilde{\omega}_1, \tilde{\omega}_2\rangle = \frac{1}{2}(\|\tilde{\omega}_1\|^2 + \|\tilde{\omega}_2\|^2) - \|\tilde{\omega}_1 - \tilde{\omega}_2\|^2.$$

According to Assumption I, it is easy to have

$$\mathbb{E}[\phi(\widetilde{x}_{j+1})] \leq \mathbb{E}[\phi(\widetilde{x}_j)] + \langle\nabla\phi(\widetilde{x}_j), \widetilde{x}_{j+1} - \widetilde{x}_j\rangle + \frac{L}{2}\|\widetilde{x}_{j+1} - \widetilde{x}_j\|^2$$

$$
\begin{aligned}
&= \mathbb{E}[\phi(\widetilde{x}_j)] + \eta'_j\langle\nabla\phi(\widetilde{x}_j), D_j\rangle + \frac{L}{2}\|\eta'_j D_j\|^2 \\
&= \mathbb{E}[\phi(\widetilde{x}_j)] - \eta'_j\langle\nabla\phi(\widetilde{x}_j), \zeta_j\rangle + \eta'_j\langle\nabla\phi(\widetilde{x}_j), B_j D_{j-1} + O_j Y_j^m\rangle + \frac{L}{2}\|\eta'_j D_j\|^2.
\end{aligned}
$$

From the formula above, the following equation holds true:

$$
\begin{aligned}
&= \mathbb{E}[\phi(\widetilde{\omega}_j)] - \eta'_j(\frac{1}{2}\|\nabla\phi(\widetilde{\omega}_j)\|^2 + \frac{1}{2}\|\zeta_j\|^2 - \|\nabla\phi(\widetilde{\omega}_j) - \zeta_j\|^2) \\
&\quad + \frac{1}{4}\eta'_j[\|\nabla\phi(\widetilde{\omega}_j) + B_j D_{j-1} - O_j Y_j^m\|^2 - \|\nabla\phi(\widetilde{\omega}_j) - (B_j D_{j-1} - O_j Y_j^m)\|^2] + \frac{L}{2}\|\eta'_j D_j\|^2 \\
&\leq \mathbb{E}[\phi(\widetilde{\omega}_j)] - \eta'_j\left(\frac{1}{2}\|\nabla\phi(\widetilde{\omega}_j)\|^2 + \frac{1}{2}\|\zeta_j\|^2 - \|\nabla\phi(\widetilde{\omega}_j) - \zeta_j\|^2\right) \\
&\quad + \frac{\eta'_j}{4}(\|\nabla\phi(\widetilde{\omega}_j)\|^2 + \|B_j D_{j-1} - O_j Y_j^m\|^2 + \|B_j D_{j-1} - O_j Y_j^m\|^2 \\
&\quad - \|\nabla\phi(\widetilde{\omega}_j)\|^2) + \frac{L}{2}\eta'^2_j(\| - \zeta_j + B_j D_{j-1} - O_j Y_j^m\|^2) \\
&\leq \mathbb{E}[\phi(\widetilde{\omega}_j)] - \eta'_j\left(\frac{1}{2}\|\nabla\phi(\widetilde{\omega}_j)\|^2 + \frac{1}{2}\|\zeta_j\|^2 - \|\nabla\phi(\widetilde{\omega}_j) - \zeta_j\|^2\right) \\
&\quad + \frac{\eta'_j}{4}(\|B_j D_{j-1}\|^2 + \|O_j Y_j^m\|^2 + \|B_j D_{j-1}\|^2 + \|O_j Y_j^m\|^2 \\
&\quad + L\eta'^2_j(\|\zeta_j\|^2 + \|B_j D_{j-1}\|^2 + \|O_j Y_j^m\|^2) \\
&= \mathbb{E}[\phi(\widetilde{\omega}_j)] - \eta'_j\left(\frac{1}{2}\|\nabla\phi(\widetilde{\omega}_j)\|^2 + \frac{1}{2}\|\zeta_j\|^2 - \|\nabla\phi(\widetilde{\omega}_j) - \zeta_j\|^2\right) \\
&\quad + \eta'_j\|B_j D_{j-1}\|^2 + L\eta'^2_j[\|\zeta_j\| + 2\|B_j D_{j-1}\|^2] \\
&\leq \mathbb{E}[\phi(\widetilde{\omega}_j)] - \eta'_j\left(\frac{1}{2}\|\nabla\phi(\widetilde{\omega}_j)\|^2 + \frac{1}{2}\|\zeta_j\|^2 - \|\nabla\phi(\widetilde{\omega}_j) - \zeta_j\|^2\right) \\
&\quad + \frac{\eta'_j}{\mu_2^2}\|D_{j-1}\|^2 + L\eta'^2_j[\|\zeta_j\|^2 + \frac{2}{\mu_2^2}\|D_{j-1}\|^2] \\
&= \mathbb{E}[\phi(\widetilde{\omega}_j)] - (\frac{\eta'_j}{2} - L\eta_j^2)\|\zeta_j\|^2 + \eta'_j\|\nabla\phi(\widetilde{\omega}_j) - \zeta_j\|^2 - \frac{\eta_j}{2}\|\nabla\phi(\widetilde{\omega}_j)\| \\
&\quad + \left(\frac{\eta'_j}{\mu_2^2} + L\eta'^2_j\frac{2}{\mu_2^2}\right)\|D_{j-1}\|^2 \\
&\leq \mathbb{E}[\phi(\widetilde{\omega}_j)] + (L\eta_{max}^2 - \frac{\eta_{min}}{2})\|\zeta_j\|^2 + \eta_{max}\|\nabla\phi(\widetilde{\omega}_j) - \zeta_j\|^2 - \frac{\eta_{min}}{2}\|\nabla\phi(\widetilde{\omega}_j)\| \\
&\quad + \left(\frac{\eta_{max}}{\mu_2^2} + L\eta_{max}^2\frac{2}{\mu_2^2}\right)\|D_{j-1}\|^2.
\end{aligned}
$$

By rearranging the formula, we derive the following result:

$$
\left\|\nabla\phi(\tilde{\omega}_j)\right\|^2 \leq \frac{2}{\eta_{\min}}\left(\mathbb{E}[\phi(\tilde{\omega}_j)] - \mathbb{E}[\phi(\tilde{\omega}_{j+1})]\right) + \frac{2\eta_{\max}}{\eta_{\min}}\left\|\nabla\phi(\tilde{\omega}_j) - \zeta_j\right\|^2
$$

$$+\left(\frac{2L\eta_{\max}^2}{\eta_{\min}}-1\right)\left\|\zeta_j\right\|^2+\frac{1}{\mu_2^2}\left(\frac{2L\eta_{\max}^2}{\eta_{\min}}+\frac{\eta_{\max}}{\eta_{\min}}\right)\left\|D_{j-1}\right\|^2.$$

Summing the inequality above from $j=0$ to m, we obtain

$$\sum_{j=0}^{m}\left\|\nabla\phi(\tilde{\omega}_j)\right\|^2\leq\frac{2}{\eta_{\min}}\left(\mathbb{E}[\phi(\tilde{\omega}_0)]-\mathbb{E}[\phi(\tilde{\omega}_*)]\right)+\frac{2\eta_{\max}}{\eta_{\min}}\sum_{j=0}^{m}\left\|\nabla\phi(\tilde{\omega}_j)-\zeta_j\right\|^2$$
$$+\left(\frac{2L\eta_{\max}^2}{\eta_{\min}}-1\right)\sum_{j=0}^{m}\left\|\zeta_j\right\|^2+\frac{2}{\mu_2^2}\sum_{j=0}^{m}\left(\frac{2L\eta_{\max}^2}{\eta_{\min}}+\frac{\eta_{\max}}{\eta_{\min}}\right)\left\|D_{j-1}\right\|^2.$$

To achieve the result above, it suffices to set the following conditions:

$$\sum_{j=0}^{m}\left\|\nabla\phi(\tilde{\omega}_j)\right\|^2\quad\leq\frac{2}{\eta_{\min}}\left(\mathbb{E}[\phi(\tilde{\omega}_0)]-\mathbb{E}[\phi(\tilde{\omega}_*)]\right)+\frac{2\eta_{\max}}{\eta_{\min}}\sum_{j=0}^{m}\left\|\nabla\phi(\tilde{\omega}_j)-\zeta_j\right\|^2$$
$$+\sum_{j=0}^{m}\left(\frac{2L\eta_{\max}^2}{\eta_{\min}}-1\right)\left\|\zeta_j\right\|^2.$$

The expected result is shown. □

Theorem 3.1. *Assuming that Assumptions I and II, and* $h_1\geq\frac{2L^2\eta_{\max}^3Nm+(N-1)(2L\eta_{\max}^2-\eta_{\min})}{2L\eta_{max}^3m}$ *hold, we obtain*

$$\mathbb{E}[\|\nabla\phi(\tilde{\omega}_m)\|^2]\leq\frac{2\lambda}{\eta_{min}(m+1)}\mathbb{E}[\phi(\tilde{\omega}_0)-\phi(\tilde{\omega}^*)].$$

Proof. It is known that

$$h_1\geq\frac{2L^2\eta_{\max}^3Nm+(N-1)(2L\eta_{\max}^2-\eta_{\min})}{2L\eta_{max}^3m}.$$

By transposing the terms, we obtain

$$\frac{2L^2\eta_{\max}^3(N-h_1)}{h_1\eta_{\min}(N-1)}m+\left(\frac{2L\eta_{\max}^2}{\eta_{\min}}-1\right)\leq 0.$$

From Lemma 3.2, it is easy to see

$$\frac{2\eta_{\max}}{\eta_{\min}}\mathbb{E}[\|\nabla\phi(\tilde{\omega}_j)-\zeta_j\|^2]\leq\frac{2L^2\eta_{\max}^3}{h_1\eta_{\min}}\left(\frac{N-h_1}{N-1}\right)\sum_{i=1}^{j}\mathbb{E}[\|\zeta_{i-1}\|^2].$$

Summing up the equation above from $j=1$ to m , then adding $\left(\frac{2L\eta_{\max}^2}{\eta_{\min}}-1\right)\sum_{j=0}^{m}\mathbb{E}[\|\zeta_j\|^2]$ to both sides of the equation , we obtain

$$\frac{2\eta_{\max}}{\eta_{\min}}\sum_{j=0}^{m}\mathbb{E}[\|\nabla\phi(\tilde{\omega}_j)-\zeta_j\|^2]+(\frac{2L\eta_{\max}^2}{\eta_{\min}}-1)\sum_{j=0}^{m}\mathbb{E}[\|\zeta_j\|^2$$

$$\leq \left(\frac{2L^2\eta_{\max}^3}{h_1\eta_{\min}}(\frac{N-h_1}{N-1})m + (\frac{2L\eta_{\max}^2}{\eta_{\min}} - 1)\right)\sum_{j=0}^{m}\mathbb{E}[\|\zeta_j\|^2] \leq 0.$$

Therefore, Lemma 3.5 can be rewritten as

$$\sum_{j=0}^{m}\mathbb{E}[\|\nabla\phi(\tilde{\omega}_j)\|^2] \leq \frac{2}{\eta_{\min}}(\mathbb{E}[\phi(\tilde{\omega}_0)] - \mathbb{E}[\phi(\tilde{\omega}_*)]).$$

Furthermore , according to the definition of ω_t in Algorithm 2, we have

$$\begin{aligned}\mathbb{E}[\|\nabla\phi(\tilde{\omega}_m)\|^2] &= \frac{1}{m+1}\sum_{j=0}^{m}\mathbb{E}[\|\nabla\phi(\tilde{\omega}_j)\|^2] \\ &\leq \frac{2}{\eta_{\min}(m+1)}(\mathbb{E}[\phi(\tilde{\omega}_0)] - \mathbb{E}[\phi(\tilde{\omega}_*)]).\end{aligned}$$

A sufficient condition for convergence is that $\frac{2}{\eta_{\min}(m+1)}(\mathbb{E}[\phi(\tilde{\omega}_0)] - \mathbb{E}[\phi(\tilde{\omega}_*)] \leq \epsilon$. A sufficient choice to satisfy the preceding inequality is $m = O\left(\frac{1}{\eta_{\min}\epsilon}\right)$. Because $\eta_{min} = b$, therefore, $m = O\left(\frac{1}{b\epsilon}\right)$. □

Theorem 3.2. *Assume that Assumptions I, and Assumptions II hold and ω_t is generated by Algorithm 2. Consider the SCGBW with $h_1 \geq \frac{2L^2\eta_{\max}^3 Nm+(N-1)(2L\eta_{\max}^2-\eta_{\min})}{2L\eta_{max}^3 m}$. Then we have*

$$\mathbb{E}[\|\nabla\phi(\omega_t)\|^2] \leq \rho^t\mathbb{E}[\|\nabla\phi(\omega_0)\|^2],$$

where $\rho = \frac{2\lambda}{\eta_{min}(m+1)} \in (0,1)$.

Proof. Given $\omega_{t-1} = \tilde{\omega}_0$ as defined in Algorithm 2, Theorem 1 can consequently be reformulated as follows:

$$\begin{aligned}\mathbb{E}[\|\nabla\phi(\omega_t)|\omega_{t-1}\|^2] &= \mathbb{E}[\|\nabla\phi(\omega_t)|\tilde{\omega}_0\|^2] \\ &\leq \frac{2}{\eta_{\min}(m+1)}(\mathbb{E}[\phi(\tilde{\omega}_0)] - \mathbb{E}[\phi(\tilde{\omega}_*)]) \\ &\leq \frac{2\lambda}{\eta_{\min}(m+1)}\|\nabla\phi(\tilde{\omega}_0)\|^2 \\ &= \frac{2\lambda}{\eta_{\min}(m+1)}\|\nabla\phi(\omega_{t-1})\|^2.\end{aligned}$$

The last inequality stems from Assumption II. Consequently, by evaluating the aforementioned expectation, we are able to determine that

$$\begin{aligned}\mathbb{E}[\|\nabla\phi(\omega_t)\|^2] &\leq \frac{2\lambda}{\eta_{\min}(m+1)}\mathbb{E}[\|\nabla\phi(\omega_{t-1})\|^2] \\ &\leq \left(\frac{2\lambda}{\eta_{\min}(m+1)}\right)^t\mathbb{E}[\|\nabla\phi(\omega_0)\|^2].\end{aligned}$$

If we let $\rho = \frac{2\lambda}{\eta_{min}(m+1)}$, the anticipated outcomes are thereby established.

To ensure the linear convergence of Algorithm 2, Theorem 2 necessitates that the parameter ρ satisfies $0 < \rho < 1$. Substituting the expression for ρ, we obtain the following critical condition on the parameters: $\rho = \frac{2\lambda}{\eta_{\min}(m+1)} < 1$. Consequently, the parameters must be selected to fulfill the inequality $\frac{2\lambda}{b(m+1)} \in (0,1)$. □

4. Numerical experiments

4.1. Overview of the research problems and datasets

Our research primarily centers on finite and minimization problems. Consequently, we assessed the SCGBW algorithm through the following two models.

Model one (Non-convex SVM model):

$$\min_{\omega\in\Re^d} \frac{1}{n}\sum_{i=1}^{n} \psi_i(\omega) + \lambda\|\omega\|^2,$$

where $\psi_i(\omega) = 1 - \tanh(b_i\langle\omega, a_i\rangle)$, $a_j \in \mathbb{R}^d$, $b_j \in \{-1, 1\}$, and $\lambda > 0$ is a regularization coefficient.

The SVM model has been extensively utilized across various machine learning domains, including image recognition, text classification, and recognition of handwritten digits. In assessing the efficiency of our algorithm, we primarily concentrate on classification tasks, which involve both text and image data.

Model two (ERM model with a non-convex sigmoid loss function):

$$\min_{\omega\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \psi_i(\omega) + \frac{\lambda}{2}\|\omega\|_2^2,$$

where $\psi_i(\omega) = 1/\left[1 + \exp\left(b_i a_i^T \omega\right)\right]$ and $\lambda > 0$.

The ERM model, particularly when equipped with a non-convex sigmoid loss function, holds significant importance in numerous practical mathematical problems, with the binary classification problem being the most exemplary. This type of problem has found extensive application across various domains. The non-convex ERM model is specifically designed to maximize classification accuracy. Additionally, it has been empirically demonstrated that the sigmoid function generally outperforms other loss functions in terms of classification performance.

The experimental evaluation utilized three benchmark datasets: Adult, IJCNN, and Covtype. The Adult dataset, derived from the 1994 US Census, is extensively cited in literature as a standard for algorithm performance comparisons. The IJCNN dataset traces its origin to a physical system and was first used to design the competition tasks for IJCNN 2001 [50]. The Covtype dataset catalogs the types of forest cover across four wilderness areas within Colorado's Roosevelt National Forest, with each instance characterized by multiple cartographic attributes; its provenance is accredited to Blackard and Dean [51]. A summary of the key statistics for these datasets is provided in Table 1.

Table 1. Test datasets.

Datasets	Training samples (n)	Dimensions (d)
Adult	32,562	123
IJCNN	49,990	22
Covtype	581,012	54

For this study, we used three canonical datasets: Adult, IJCNN, and Covtype. Detailed information regarding these datasets is provided in Table 1. To facilitate a fair and standardized comparison of our

algorithm with existing methods, we consistently utilized a batch size $S = \sqrt{N}$ across all experiments, with N denoting the training sample's size. This approach ensures that comparisons are conducted under uniform conditions. Moreover, we initialized our algorithm with a zero vector.

4.2. Numerical results

This section focuses on exploring the potential impacts of parameters such as B, b, μ_1, μ_2, and the initial step size η on the algorithm's performance. Meanwhile, to highlight the superiority of our method, we conducted comparative analyses of the SCGBW algorithm with six other classical algorithms, namely SVRG, SGD, SAGA, SARAH, Adam, and RMSProp, under the same conditions.

4.2.1. Influence of the initial step size η in SCGBW

We investigate the effect of the initial step size η on the SCGBW algorithm, with parameters $B = 2$, $b = 0.1$, $\mu_1 = 0.1$, $\mu_2 = 10$, and $\lambda = 10^{-3}$. The numerical performance of the SCGBW algorithm is relatively robust with respect to the choice of the initial step size η. Specifically, even when the initial step size η is subject to substantial variation, the algorithm can still achieve a certain degree of performance improvement. This subsection provides empirical evidence to support this observation. Figures 1 and 2, respectively, verify the efficacy of the earlier assertion when solving Model One and Model Two on the three datasets described in Section 4.1.

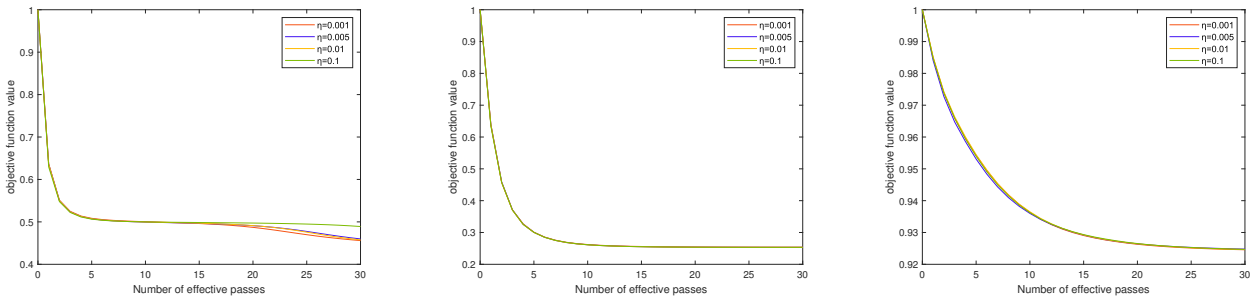


Figure 1. Impact of the initial step size η on Model One for the Adult, IJCNN, and Covtype datasets.

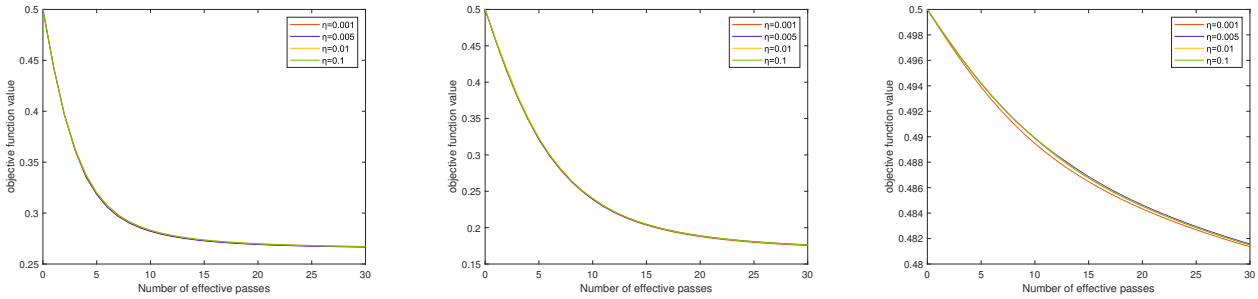


Figure 2. Impact of the initial step size η on Model Two for the Adult, IJCNN, and Covtype datasets.

4.2.2. Influence of the parameters B and b in SCGBW

The outcomes of the SCGBW algorithm, which uses diverse values of B and b for determining the step size via the bandwidth-based approach, are depicted in Figures 3–6. As illustrated in these figures, while the initial choices of B and b exert some influence on SCGBW's performance, the convergence behavior remains largely unaffected. Consequently, we opted for $B = 2$; $b = 0.1$.

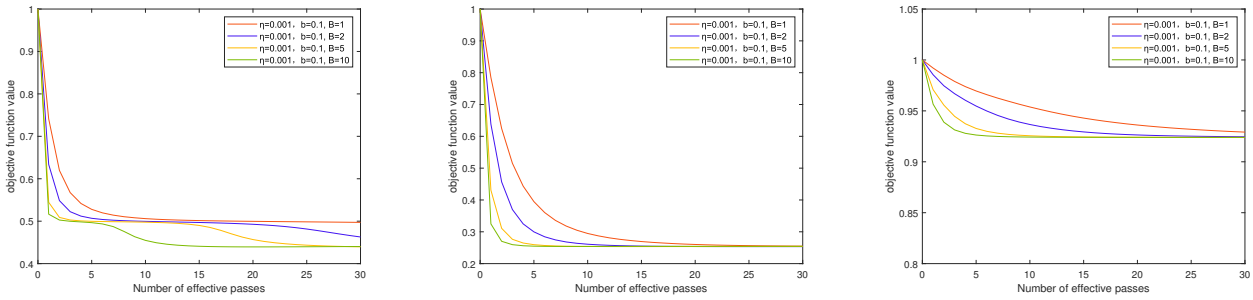


Figure 3. Impact of the parameter B on Model One for the Adult, IJCNN, and Covtype datasets.

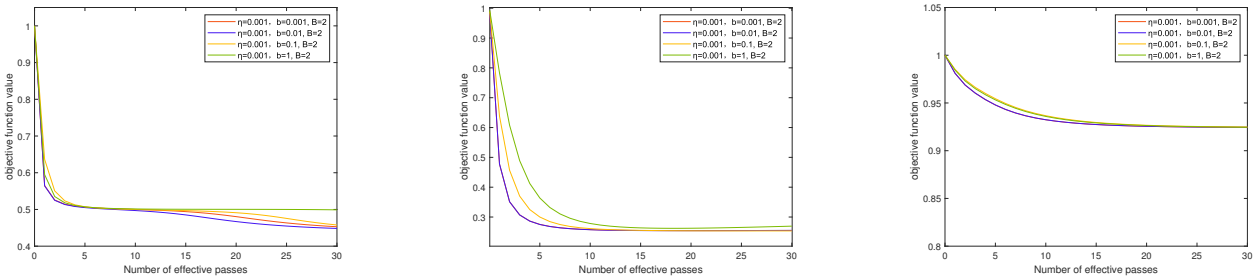


Figure 4. Impact of the parameter b on Model One for the Adult, IJCNN, and Covtype datasets.

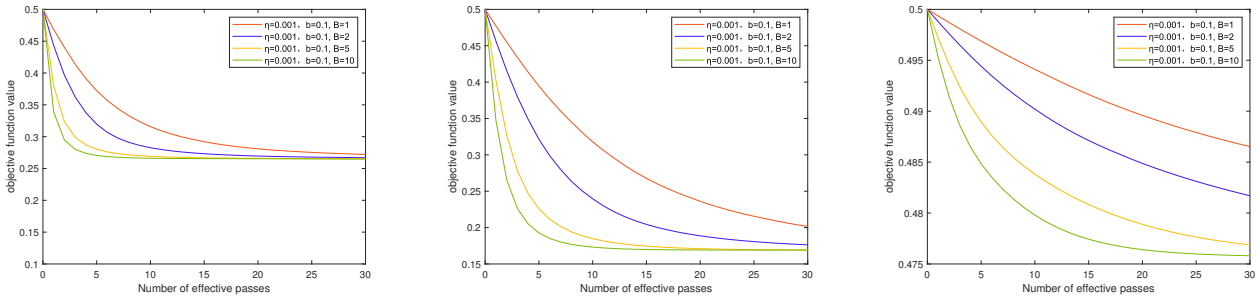


Figure 5. Impact of the parameter B on Model Two for the Adult, IJCNN, and Covtype datasets.

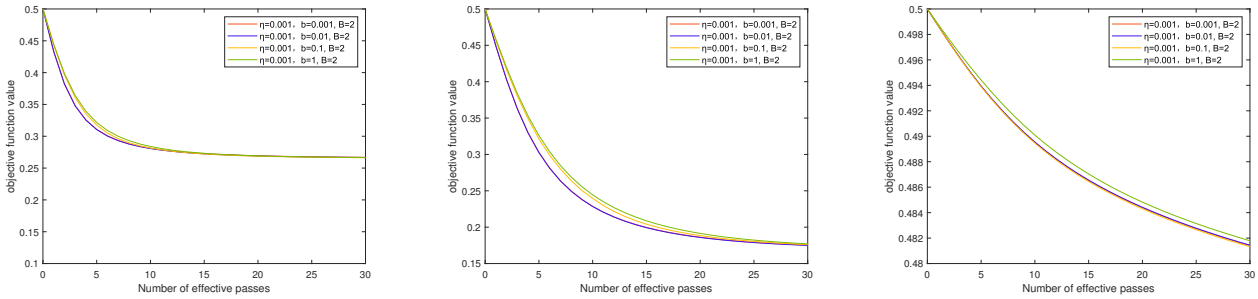


Figure 6. Impact of the parameter b on Model Two for the Adult, IJCNN, and Covtype datasets.

4.2.3. Influence of the parameters μ_1 and μ_2 in SCGBW

In Figures 7–10 , we tested the influence of the parameters $\mu 1$ and $\mu 2$ in the three-term conjugate direction on the algorithm. In these figures, we set $\eta = 0.001$, $b = 0.1$, and $B = 0.2$, with $\mu 1$ taking values of 0.01, 0.1, 1, 2, and $\mu 2$ taking values of 1, 5, 10, and 100. It can be observed that the parameters mu_1 and μ_2 have little impact on the algorithm. Therefore, this paper selects $\mu_1 = 0.1$ and $\mu 2 = 10$.

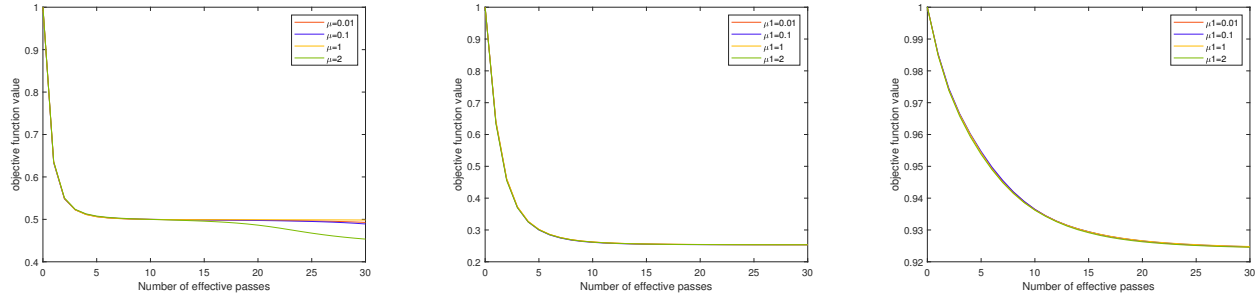


Figure 7. Impact of the parameter μ_1 on Model One for the Adult, IJCNN, and Covtype datasets.

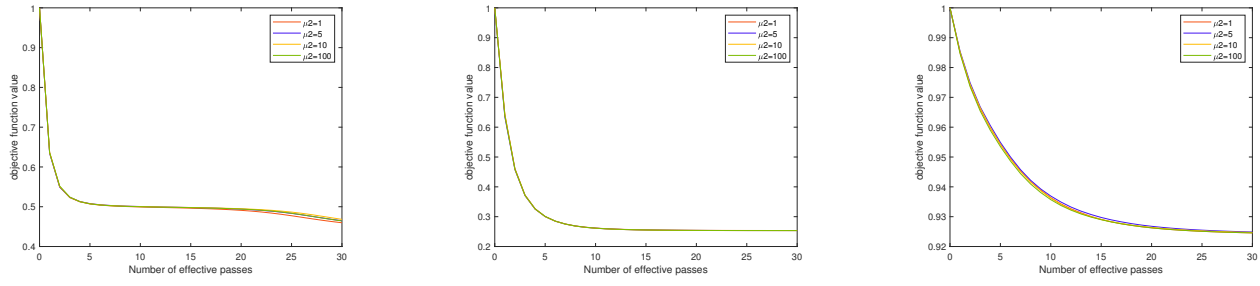


Figure 8. Impact of the parameter μ_2 on Model One for the Adult, IJCNN, and Covtype datasets.

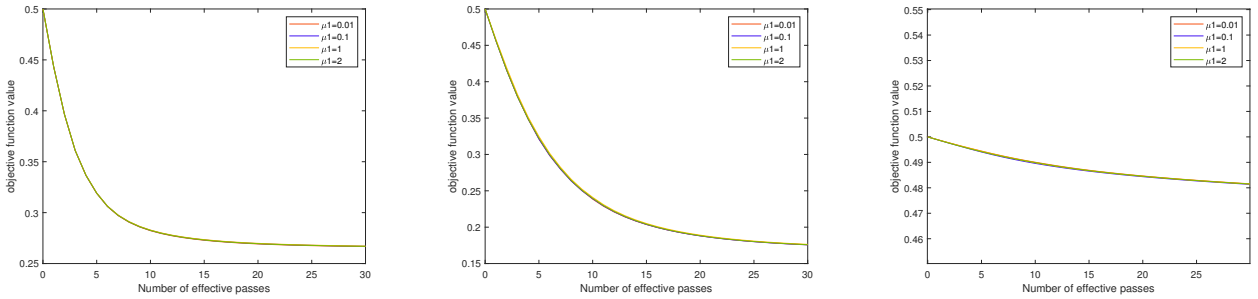


Figure 9. Impact of the parameter μ_1 on Model Two for the Adult, IJCNN, and Covtype datasets.

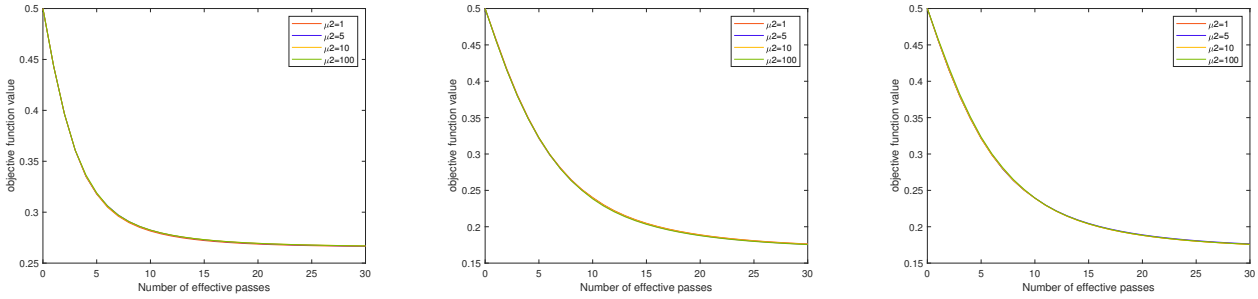
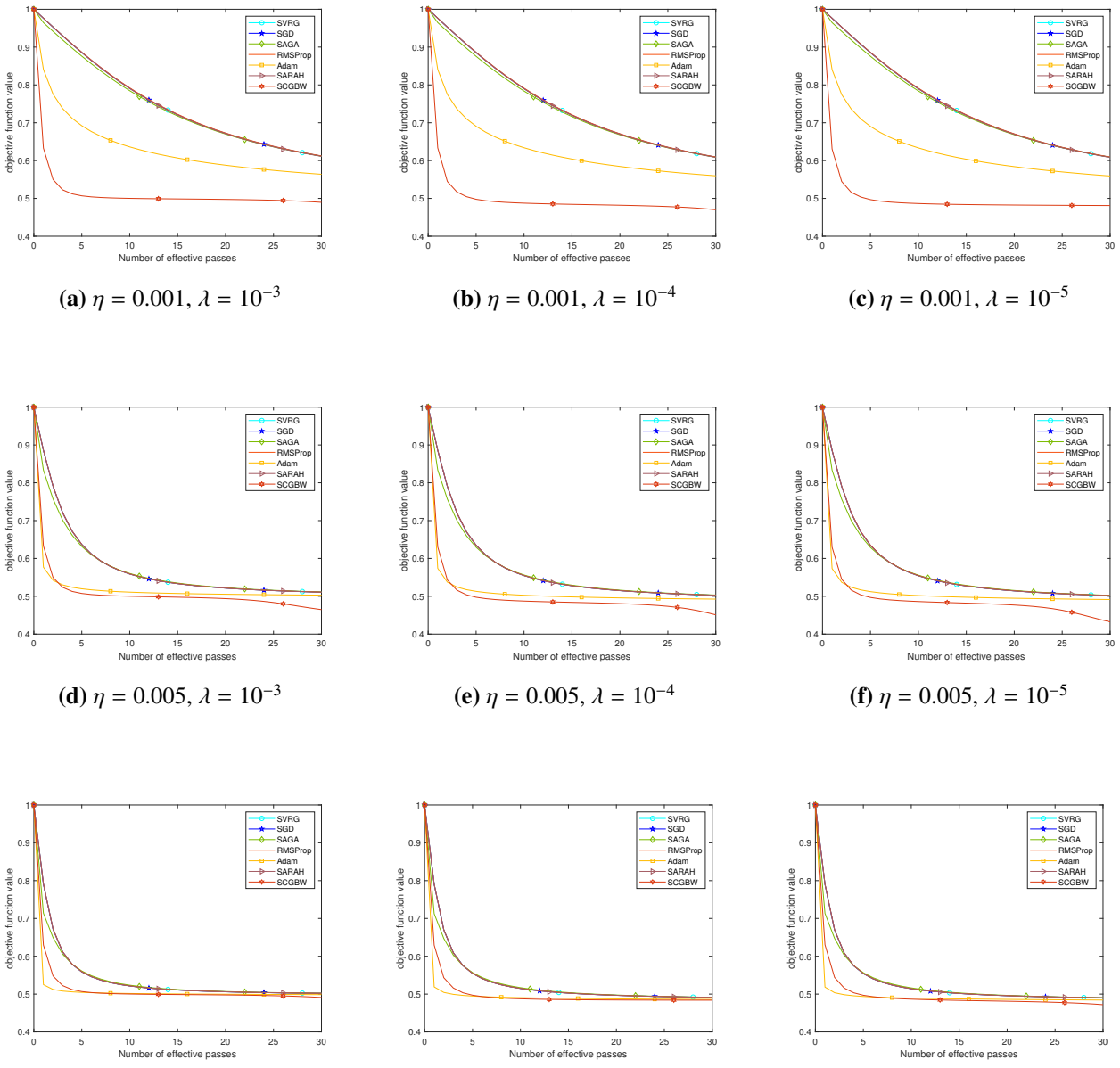


Figure 10. Impact of the parameter μ_2 on Model Two for the Adult, IJCNN, and Covtype datasets.

4.2.4. Performance of the SCGBW for Models One and Two

A pivotal aspect of our experimental design is the step size strategy, where a constant value η is used for all algorithms under comparison, including SVRG, SGD, SAGA, Adam, SARAH, and RMSProp. These methods can be categorized into two groups: The first three are established stochastic algorithms, and the remaining three are widely adopted adaptive optimizers. To ensure equitable comparisons, we maintained consistent initial step size η and regularization parameter λ configurations across all algorithms within each experimental setting. Specifically, we sequentially assigned η values of 0.001, 0.005, and 0.01, while λ was tested at $10^{-3}, 10^{-4}$, and 10^{-5}, which are the consistent with common practices in the research literature. The Adam algorithm was configured with hyperparameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Each figure presents a 3×3 grid of subplots. Notably, subplots within the same row share identical η values, while those within the same column maintain consistent λ values. For instance, the top row of Figure 11 maintains a fixed η while demonstrating λ values of $10^{-3}, 10^{-4}$, and 10^{-5} from left to right and sets $m = N^{1/2}$. To enhance visual discrimination between highly overlapping algorithm trajectories, we used distinct markers at different iteration intervals. From Figures 11–16, we can observe that when the initial step size is small, the proposed SCGBW algorithm reaches the steady state faster than the classical random algorithms (SVRG, SGD, SAGA, Adam, SARAH, and RMSProp) on Model One and Model Two, which means it has a higher convergence rate. In the theoretical analysis,

it is shown that for the proposed algorithm SCGBW, a larger inner loop size m leads to a faster descent speed. Figure 17 tests the descent speed of SCGBW with different values of m on Model One using the Adult dataset, providing empirical results that support the theoretical analysis. Moreover, as shown in Figure 17, the descent speed of the algorithm when $m = N^{3/5}$ is better than that when $m = N^{1/2}$, further demonstrating that the proposed SCGBW algorithm outperforms the other classical algorithms.

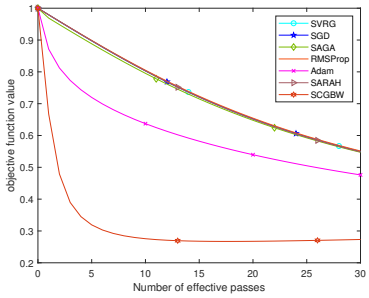


(a) $\eta = 0.001$, $\lambda = 10^{-3}$ **(b)** $\eta = 0.001$, $\lambda = 10^{-4}$ **(c)** $\eta = 0.001$, $\lambda = 10^{-5}$

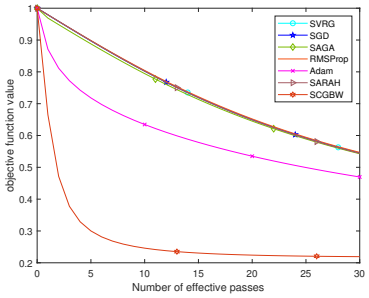
(d) $\eta = 0.005$, $\lambda = 10^{-3}$ **(e)** $\eta = 0.005$, $\lambda = 10^{-4}$ **(f)** $\eta = 0.005$, $\lambda = 10^{-5}$

(g) $\eta = 0.01$, $\lambda = 10^{-3}$ **(h)** $\eta = 0.01$, $\lambda = 10^{-4}$ **(i)** $\eta = 0.01$, $\lambda = 10^{-5}$

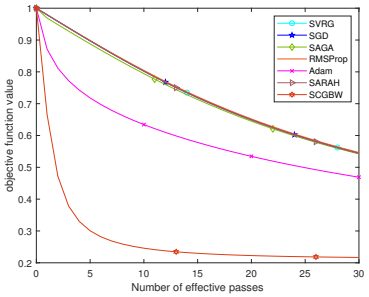
Figure 11. Comparative evaluation of the seven algorithms on the Adult dataset for Model One.



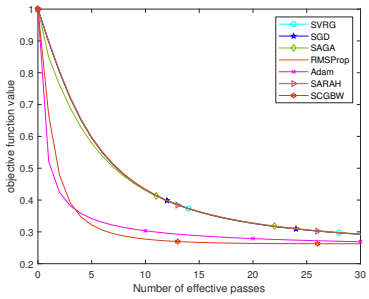
(a) $\eta = 0.001$, $\lambda = 10^{-3}$



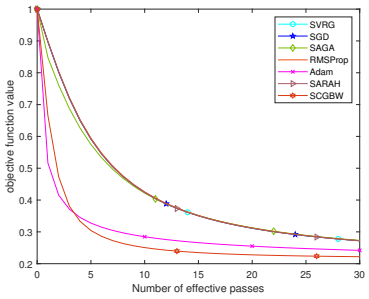
(b) $\eta = 0.001$, $\lambda = 10^{-4}$



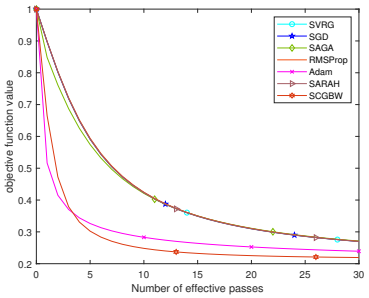
(c) $\eta = 0.001$, $\lambda = 10^{-5}$



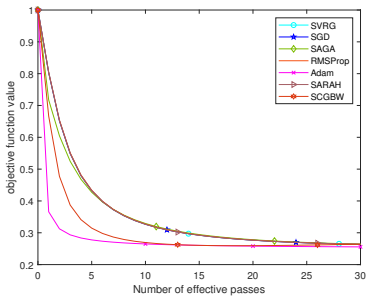
(d) $\eta = 0.005$, $\lambda = 10^{-3}$



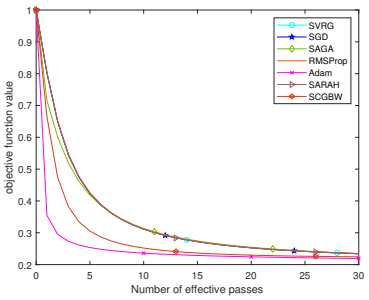
(e) $\eta = 0.005$, $\lambda = 10^{-4}$



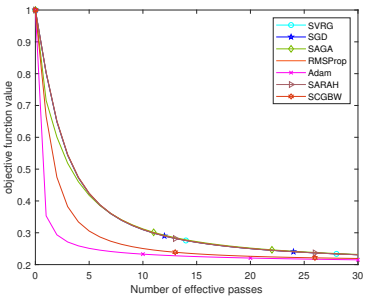
(f) $\eta = 0.005$, $\lambda = 10^{-5}$



(g) $\eta = 0.01$, $\lambda = 10^{-3}$

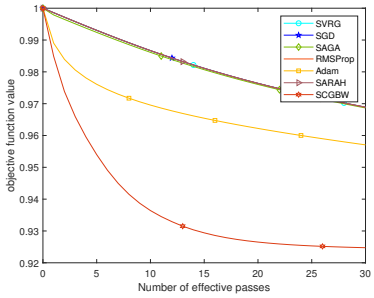


(h) $\eta = 0.01$, $\lambda = 10^{-4}$

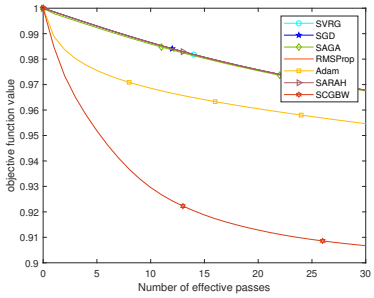


(i) $\eta = 0.01$, $\lambda = 10^{-5}$

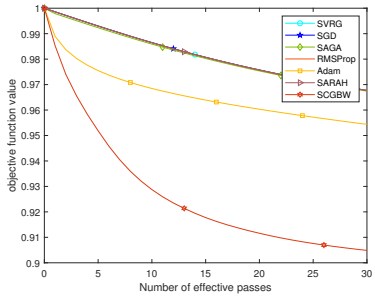
Figure 12. Comparative evaluation of the seven algorithms on the IJCNN dataset for Model One.



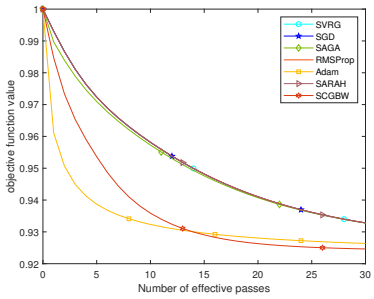
(a) $\eta = 0.001$, $\lambda = 10^{-3}$



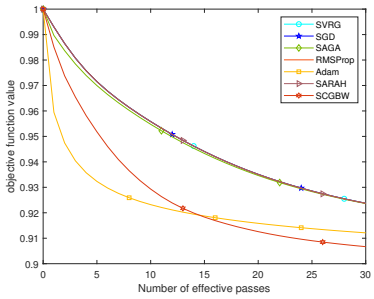
(b) $\eta = 0.001$, $\lambda = 10^{-4}$



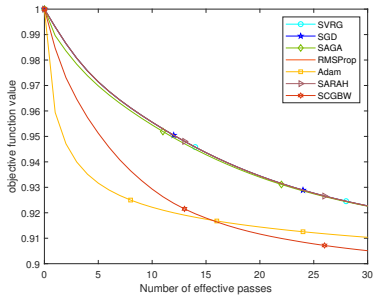
(c) $\eta = 0.001$, $\lambda = 10^{-5}$



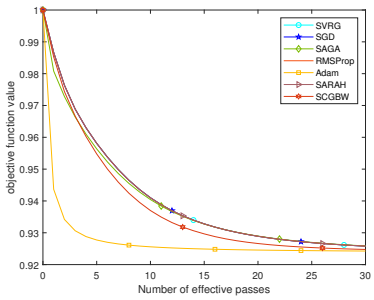
(d) $\eta = 0.005$, $\lambda = 10^{-3}$



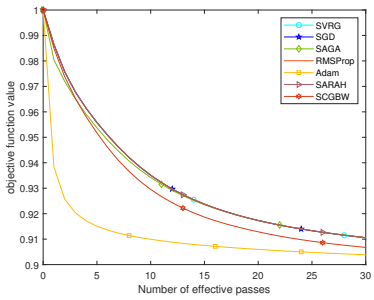
(e) $\eta = 0.005$, $\lambda = 10^{-4}$



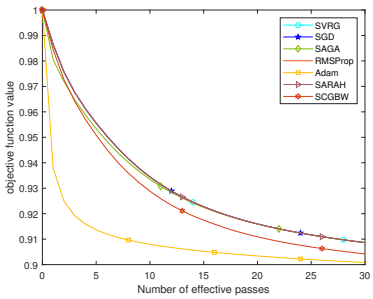
(f) $\eta = 0.005$, $\lambda = 10^{-5}$



(g) $\eta = 0.01$, $\lambda = 10^{-3}$

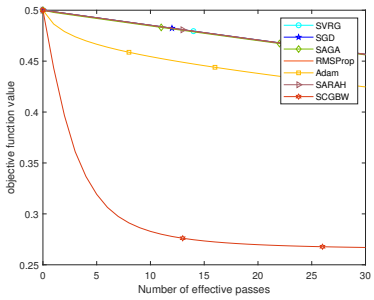


(h) $\eta = 0.01$, $\lambda = 10^{-4}$

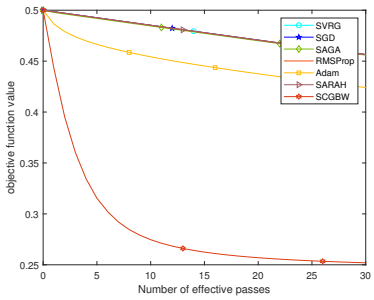


(i) $\eta = 0.01$, $\lambda = 10^{-5}$

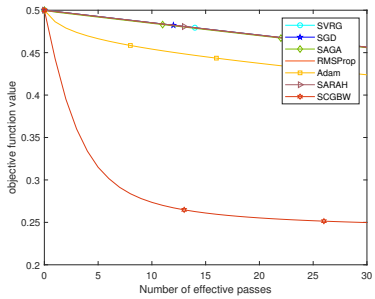
Figure 13. Comparative evaluation of the seven algorithms on the Covtype dataset for Model One.



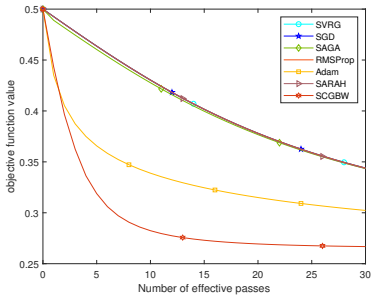
(a) $\eta = 0.001$, $\lambda = 10^{-3}$



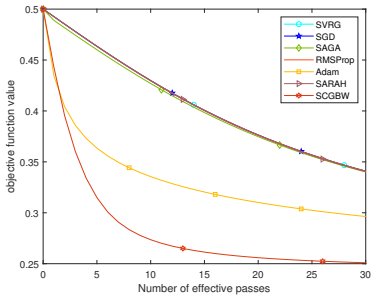
(b) $\eta = 0.001$, $\lambda = 10^{-4}$



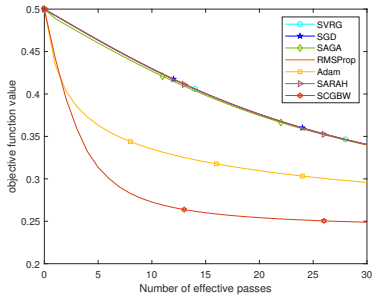
(c) $\eta = 0.001$, $\lambda = 10^{-5}$



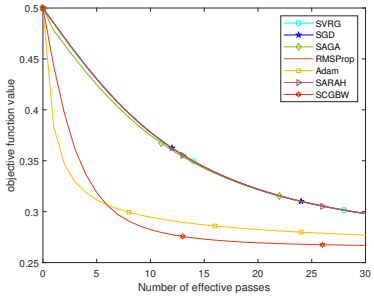
(d) $\eta = 0.005$, $\lambda = 10^{-3}$



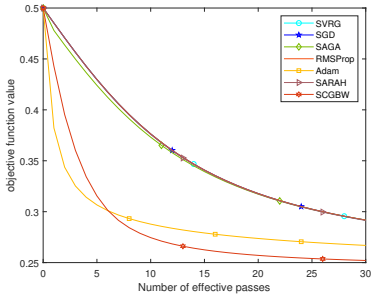
(e) $\eta = 0.005$, $\lambda = 10^{-4}$



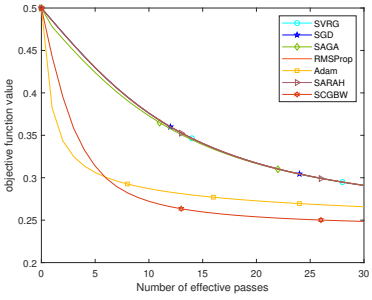
(f) $\eta = 0.005$, $\lambda = 10^{-5}$



(g) $\eta = 0.01$, $\lambda = 10^{-3}$

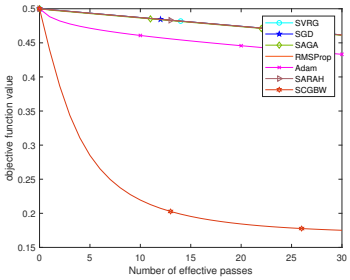


(h) $\eta = 0.01$, $\lambda = 10^{-4}$

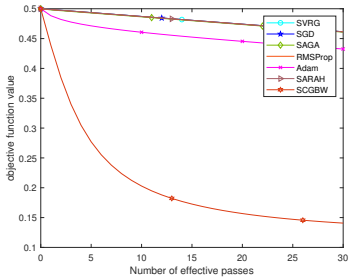


(i) $\eta = 0.01$, $\lambda = 10^{-5}$

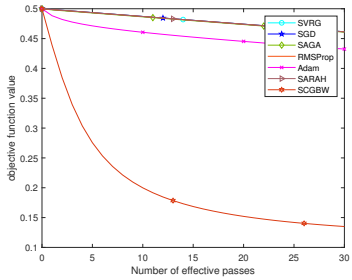
Figure 14. Comparative evaluation of the seven algorithms on the Adult dataset for Model Two.



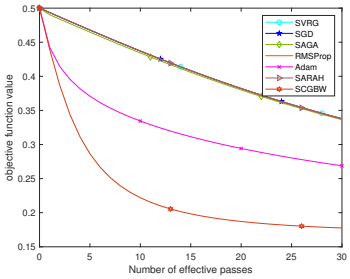
(a) $\eta = 0.001$, $\lambda = 10^{-3}$



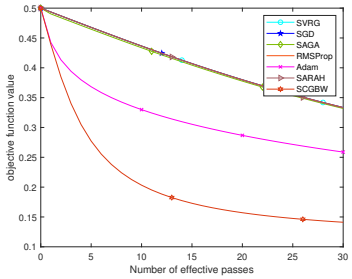
(b) $\eta = 0.001$, $\lambda = 10^{-4}$



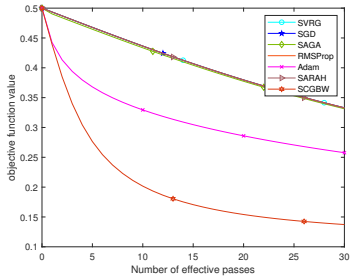
(c) $\eta = 0.001$, $\lambda = 10^{-5}$



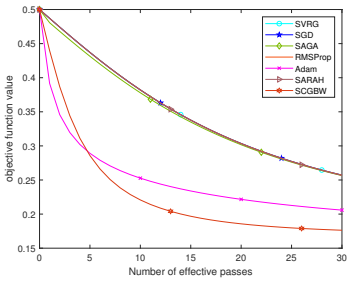
(d) $\eta = 0.005$, $\lambda = 10^{-3}$



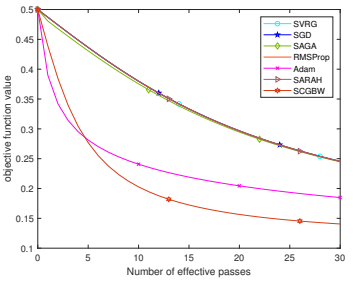
(e) $\eta = 0.005$, $\lambda = 10^{-4}$



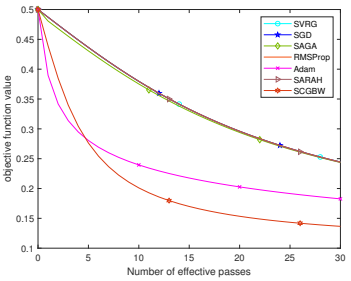
(f) $\eta = 0.005$, $\lambda = 10^{-5}$



(g) $\eta = 0.01$, $\lambda = 10^{-3}$

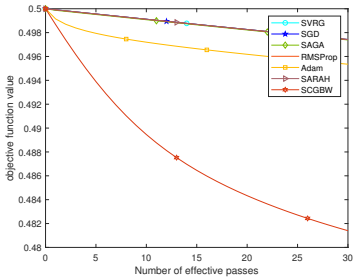


(h) $\eta = 0.01$, $\lambda = 10^{-4}$

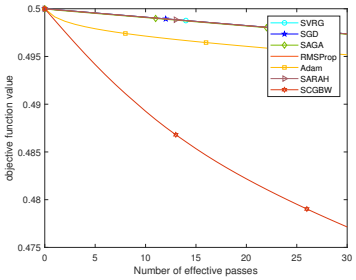


(i) $\eta = 0.01$, $\lambda = 10^{-5}$

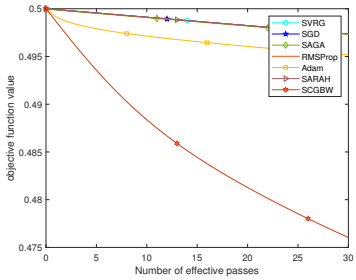
Figure 15. Comparative evaluation of the seven algorithms on the IJCNN dataset for Model Two.



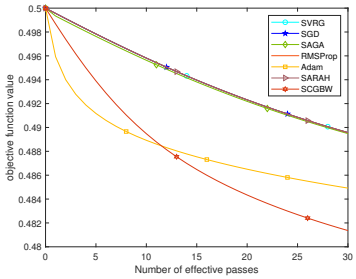
(a) $\eta = 0.001$, $\lambda = 10^{-3}$



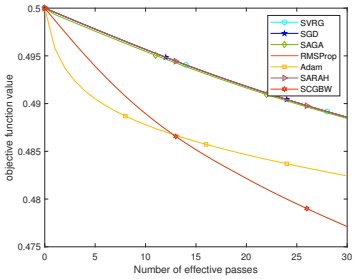
(b) $\eta = 0.001$, $\lambda = 10^{-4}$



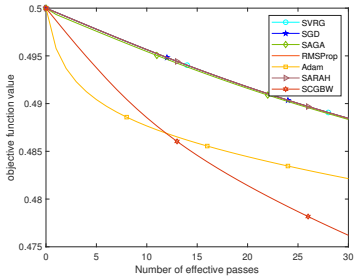
(c) $\eta = 0.001$, $\lambda = 10^{-5}$



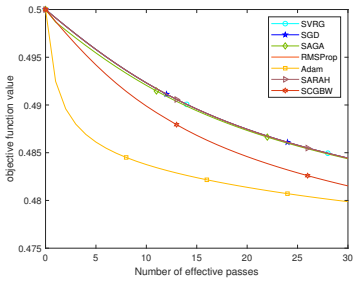
(d) $\eta = 0.005$, $\lambda = 10^{-3}$



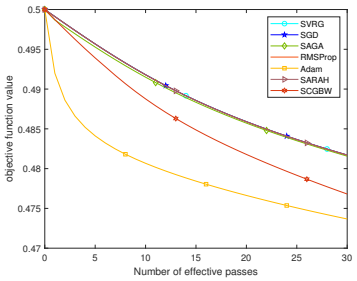
(e) $\eta = 0.005$, $\lambda = 10^{-4}$



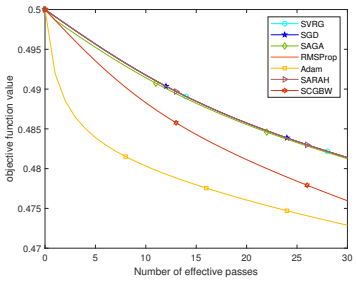
(f) $\eta = 0.005$, $\lambda = 10^{-5}$



(g) $\eta = 0.01$, $\lambda = 10^{-3}$

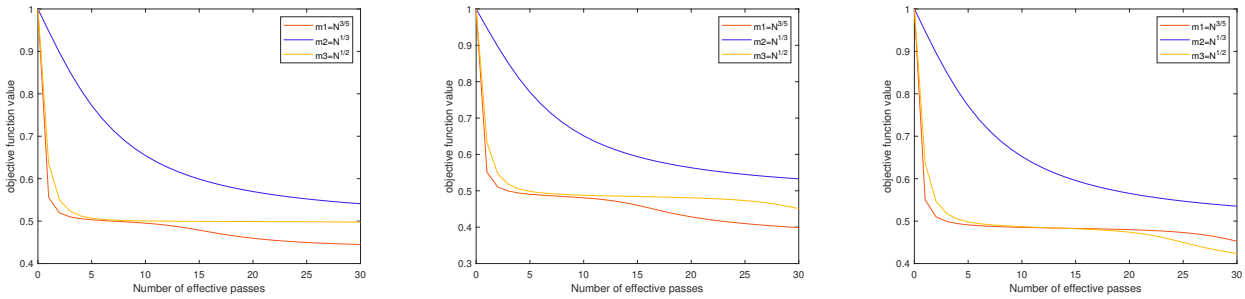


(h) $\eta = 0.01$, $\lambda = 10^{-4}$



(i) $\eta = 0.01$, $\lambda = 10^{-5}$

Figure 16. Comparative evaluation of the seven algorithms on the Covtype dataset for Model Two.



(a) $\lambda = 0.001$ **(b)** $\lambda = 0.0001$ **(c)** $\lambda = 0.00001$

Figure 17. Impact of different parameter values of the m on Model One for the Adult detaset.

5. Conclusions

This paper has proposed a biased stochastic three-term CG algorithm with a bandwidth-based step size (SCGBW) for addressing non-convex stochastic optimization problems. The core innovation lies in the integration of the SARAH gradient estimator with a modern three-term CG framework, enhanced by an adaptive bandwidth-based step size strategy. A pivotal element of our approach is the modification of the conventional gradient difference term y_j to y_j^m, which incorporates both gradient and function value information.

Through a rigorous theoretical analysis under gradient Lipschitz continuity and λ–gradient dominated assumptions, we have established not only the almost sure convergence of SCGBW to the global optima but also demonstrated its linear convergence rate for a class of non-convex functions. This theoretical guarantee addresses a fundamental challenge in stochastic optimization by ensuring both stability and rapid convergence.

Comprehensive numerical experiments on established machine learning benchmarks, including non-convex support vector machines and empirical risk minimization problems, substantiate the efficacy of our approach. The results consistently demonstrate that SCGBW outperforms widely used optimization algorithms such as SGD, SVRG, SAGA, SARAH, Adam, and RMSProp across multiple datasets including the Adult, IJCNN, and Covtype datasets. The algorithm exhibits several notable advantages: (i) Robustness to the initial step size selection, maintaining performance across varying configurations; (ii) accelerated convergence rates, which are particularly evident when initial step sizes are modest; and (iii) enhanced stability during training iterations, leading to superior final solution quality.

Future research directions will focus on: (i) developing more sophisticated step size adaptation mechanisms to achieve faster descent speeds with larger initial step sizes; (ii) investigating novel strategies for preserving Hessian positive definiteness independent of traditional modification schemes, similar to the approach explored by Wang et al. [52], to achieve even more robust search directions; (iii) integrating advanced variance reduction techniques to further enhance computational efficiency; (iv) extending the framework to distributed and large-scale learning environments, and (v) applying and rigorously evaluating the algorithm across a broader spectrum of deep learning architectures.

Use of AI tools declaration

The authors declare they have not used artificial intelligence (AI) tools in the creation of this article.

Acknowledgments

This work is supported by the "Strengthening Foundations" Action Plan (Guangxi Basic Research Plan) Project (Grant No. 2026GXNSFDA00640003), the Guangxi Key Technologies R&D Program (Grant No. FN2504240016, FN2504240023, and AB25069188), the Guangxi Science and Technology Base and Talent Project (Grant No. AD22080047), and the special foundation for Guangxi Ba Gui Scholars (Grant No. GXR-6BG2424008).

Conflict of interest

The authors declare there are no conflicts of interest.

References

1. M. A. Cauchy, Méthode générale pour la résolution des systemes déquations simultanées, *C. R. Hebd. Seances Acad. Sci.*, **25** (1874), 536–538.
2. L. Bottou, Large-scale machine learning with stochastic gradient descent, in *Proceedings of COMPSTAT's 2010*, (2010), 177–186. https://doi.org/10.1007/978-3-7908-2604-3_16
3. J. Duchi, E. Hazan, Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, *J. Mach. Learn. Res.*, **12** (2011), 2121–2159.
4. T. Tieleman, G. Hinton, Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning, in *Technical Report*, University of Toronto, 2017.
5. D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, preprint, arXiv:1412.6980. https://doi.org/10.48550/arXiv.1412.6980
6. T. Dozat, Incorporating Nesterov momentum into Adam, in *4th International Conference on Learning Representations*, ICLR Workshop Track, 2016.
7. N. Roux, M. Schmidt, F. Bach, A stochastic gradient method with an exponential convergence rate for finite training sets, *Adv. Neural Inf. Process. Syst.*, **25** (2012).
8. A. Defazio, F. Bach, S. Lacoste-Julien, Saga: A fast incremental gradient method with support for non-strongly convex composite objectives, *Adv. Neural Inf. Process. Syst.*, **27** (2014).
9. R. Johnson, T. Zhang, Accelerating stochastic gradient descent using predictive variance reduction, *Adv. Neural Inf. Process. Syst.*, **26** (2013).
10. L. M. Nguyen, J. Liu, K. Scheinberg, M. Takáč, SARAH: A novel method for machine learning problems using stochastic recursive gradient, in *International Conference on Machine Learning*, PMLR, (2017), 2613–2621.
11. M. R. Hestenes, E. Stiefel, Methods of conjugate gradients for solving linear systems, *J. Res. Natl. Bur. Stand.*, **49** (1952), 409–436. https://doi.org/10.6028/jres.049.044

12. R. Fletcher, C. M. Reeves, Function minimization by conjugate gradients, *Comput. J.*, **7** (1964), 149–154. https://doi.org/10.1093/comjnl/7.2.149

13. Y. H. Dai, Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, *SIAM J. Optim.*, **10** (1999), 177–182. https://doi.org/10.1137/S1052623497318992

14. B. T. Polyak, The conjugate gradient method in extremal problems, *USSR Comput. Math. Math. Phys.*, **9** (1969), 94–112. https://doi.org/10.1016/0041-5553(69)90035-4

15. E. Polak, G. Ribiere, Note sur la convergence de méthodes de directions conjuguées, *Rev. Fr. d'informatique Rech. Oper. Ser. Rouge*, **3** (1969), 35–43. https://doi.org/10.1051/m2an/196903R100351

16. N. Andrei, Another hybrid conjugate gradient algorithm for unconstrained optimization, *Numerical Algorithms*, **47** (2008), 143–156. https://doi.org/10.1007/s11075-007-9152-9

17. M. Raydan, The barzilai and borwein gradient method for the large scale unconstrained minimization problem, *SIAM J. Optim.*, **7** (1997), 26–33. https://doi.org/10.1137/S1052623494266365

18. G. Yu, L. Guan, W. Chen, Spectral conjugate gradient methods with sufficient descent property for large-scale unconstrained optimization, *Optim. Methods Software*, **23** (2008), 275–293. https://doi.org/10.1080/10556780701661344

19. E. G. Birgin, J. M. Martínez, A spectral conjugate gradient method for unconstrained optimization, *Appl. Math. Optim.*, **43** (2001), 117–128. https://doi.org/10.1007/s00245-001-0003-0

20. L. Zhang, W. Zhou, D. Li, Some descent three-term conjugate gradient methods and their global convergence, *Optim. Methods Software*, **22** (2007), 697–711. https://doi.org/10.1080/10556780701223293

21. N. N. Schraudolph, T. Graepel, Combining conjugate direction methods with stochastic approximation of gradients, in *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, PMLR, (2003), 248–253.

22. H. Jiang, P. Wilford, A stochastic conjugate gradient method for the approximation of functions, *J. Comput. Appl. Math.*, **236** (2012), 2529–2544. https://doi.org/10.1016/j.cam.2011.12.012

23. R. H. Byrd, G. M. Chin, J. Nocedal, Y. Wu, Sample size selection in optimization methods for machine learning, *Math. Program.*, **134** (2012), 127–155. https://doi.org/10.1007/s10107-012-0572-5

24. X. B. Jin, X. Y. Zhang, K. Huang, G. G. Geng, Stochastic conjugate gradient algorithm with variance reduction, *IEEE Trans. Neural Networks Learn. Syst.*, **30** (2018), 1360–1369. https://doi.org/10.1109/TNNLS.2018.2885887

25. K. A. Alnowibet, S. Mahdi, A. M. Alshamrani, K. M. Sallam, A. W. Mohamed, A family of hybrid stochastic conjugate gradient algorithms for local and global minimization problems, *Mathematics*, **10** (2022), 3575. https://doi.org/10.3390/math10193595

26. G. Yuan, J. Lu, Z. Jin, J. Yu, A novel stochastic conjugate gradient algorithm based on a stochastic differential equation perspective, *Eur. J. Oper. Res.*, **332** (2026), 505–521. https://doi.org/10.1016/j.ejor.2026.02.034

27. C. Ouyang, C. Lu, X. Zhao, R. Huang, G. Yuan, Y. Jiang, Stochastic three-term conjugate gradient method with variance technique for non-convex learning, *Stat. Comput.*, **34** (2024), 107. https://doi.org/10.1007/s11222-024-10409-5

28. J. Barzilai, J. M. Borwein, Two-point step size gradient methods, *IMA J. Numer. Anal.*, **8** (1988), 141–148. https://doi.org/10.1093/imanum/8.1.141
29. X. Wang, Y. X. Yuan, On the convergence of stochastic gradient descent with bandwidth-based step size, *J. Mach. Learn. Res.*, **24** (2023), 1–49.
30. Z. Wei, G. Yu, G. Yuan, Z. Lian, The superlinear convergence of a modified bfgs-type method for unconstrained optimization, *Comput. Optim. Appl.*, **29** (2004), 315–332. https://doi.org/10.1023/B:COAP.0000044184.25410.39
31. G. Yuan, Z. Wei, Convergence analysis of a modified bfgs method on convex minimizations, *Comput. Optim. Appl.*, **47** (2010), 237–255. https://doi.org/10.1007/s10589-008-9219-0
32. Z. Mo, C. Ouyang, H. Pham, G. Yuan, A stochastic recursive gradient algorithm with inertial extrapolation for non-convex problems and machine learning, *Int. J. Mach. Learn. Cybern.*, **16** (2025), 4545–4559. https://doi.org/10.1007/s13042-024-02524-6
33. L. M. Nguyen, K. Scheinberg, M. Takáč, Inexact sarah algorithm for stochastic optimization, *Optim. Methods Software*, **36** (2021), 237–258. https://doi.org/10.1080/10556788.2020.1818081
34. Z. Yang, On the step size selection in variance-reduced algorithm for nonconvex optimization, *Expert Syst. Appl.*, **169** (2021), 114336. https://doi.org/10.1016/j.eswa.2020.114336
35. Z. Yang, Z. Chen, C. Wang, Accelerating mini-batch SARAH by step size rules, *Inf. Sci.*, **558** (2021), 157–173. https://doi.org/10.1016/j.ins.2020.12.075
36. H. Kim, C. Wang, H. Byun, W. Hu, S. Kim, Q. Jiao, et al., Variable three-term conjugate gradient method for training artificial neural networks, *Neural Networks*, **159** (2023), 125–136. https://doi.org/10.1016/j.neunet.2022.12.001
37. S. J. Reddi, S. Kale, S. Kumar, On the convergence of adam and beyond, preprint, arXiv:1904.09237. https://doi.org/10.48550/arXiv.1904.09237
38. J. Zhuang, T. Tang, Y. Ding, S. C. Tatikonda, N. Dvornek, X. Papademetris, et al., Adabelief optimizer: Adapting stepsizes by the belief in observed gradients, *Adv. Neural Inf. Process. Syst.*, **33**, 18795–18806.
39. C. Kou, H. Yang, A mini-batch stochastic conjugate gradient algorithm with variance reduction, *J. Global Optim.*, **87** (2023), 1009–1025. https://doi.org/10.1007/s10898-022-01205-4
40. Z. Yang, Adaptive stochastic conjugate gradient for machine learning, *Expert Syst. Appl.*, **206** (2022), 117719. https://doi.org/10.1016/j.eswa.2022.117719
41. R. Huang, Y. Qin, K. Liu, G. Yuan, Biased stochastic conjugate gradient algorithm with adaptive step size for nonconvex problems, *Expert Syst. Appl.*, **238** (2024), 121556. https://doi.org/10.1016/j.eswa.2023.121556
42. S. Ghadimi, G. Lan, Stochastic first-and zeroth-order methods for nonconvex stochastic programming, *SIAM J. Optim.*, **23** (2013), 2341–2368. https://doi.org/10.1137/120880811
43. R. M. Gower, N. Loizou, X. Qian, A. Sailanbayev, E. Shulgin, P. Richtárik, Sgd: General analysis and improved rates, in *International Conference on Machine Learning*, (2019), 5200–5209.
44. A. Rakhlin, O. Shamir, K. Sridharan, Making gradient descent optimal for strongly convex stochastic optimization, preprint, arXiv:1109.5647. https://doi.org/10.48550/arXiv.1109.5647

45. X. Wang, S. Magnússon, M. Johansson, On the convergence of step decay step-size for stochastic optimization, *Adv. Neural Inf. Process. Syst.*, **34** (2021), 14226–14238.

46. S. J. Reddi, A. Hefny, S. Sra, B. Poczos, A. Smola, Stochastic variance reduction for nonconvex optimization, in *International Conference on Machine Learning*, PMLR, (2016), 314–323.

47. L. M. Nguyen, P. H. Nguyen, P. Richtárik, K. Scheinberg, M. Takáč, M. van Dijk, New convergence aspects of stochastic gradient algorithms, *J. Mach. Learn. Res.*, **20** (2019), 1–49.

48. B. T. Polyak, Gradient methods for the minimisation of functionals, *USSR Comput. Math. Math. Phys.*, **3** (1963), 864–878. https://doi.org/10.1016/0041-5553(63)90382-3

49. Y. Nesterov, B. T. Polyak, Cubic regularization of newton method and its global performance, *Math. Program.*, **108** (2006), 177–205. https://doi.org/10.1007/s10107-006-0706-8

50. D. Prokhorov, IJCNN 2001 Neural Network Competition, in *Slide Presentation in IJCNN*, **1** (2001), 38.

51. J. A. Blackard, D. J. Dean, Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables, *Comput. Electron. Agric.*, **24** (1999), 131–151. https://doi.org/10.1016/S0168-1699(99)00046-0

52. Y. Wang, C. Ouyang, B. Hu, G. Yuan, Y. Wang, Stochastic conjugate gradient algorithm with an inertial extrapolation step for nonconvex optimization in machine learning, *J. Appl. Math. Comput.*, **72** (2026), 1–15. https://doi.org/10.5771/2193-0147-2026-1-72

AIMS Press