A Variant of BB Method with Adaptive Learning Rate for Stochastic Optimization Algorithms

Weijuan Shi, Adibah Shuib, Zuraida Alwadood

***Abstract*—With the increasing importance of machine learning, research on optimization algorithms has become increasingly vital. The basic approach involves translating a machine learning problem into an optimization problem and then solving it. First-order stochastic optimization is a key method for addressing machine learning problems. The Barzilai-Borwein (BB) step size has been widely studied, but it suffers from potential issues where the denominator approaches zero or becomes negative, leading to instability in optimization algorithms. To address these challenges, this paper proposes a new step size strategy called the Cumulative Stabilized Barzilai-Borwein (CSBB) Method. Two experiments were conducted to illustrate the variations in the CSBB step size and the evolution of gradient norms. Another two experiments, neural network training and binary classification, were conducted to evaluate the effectiveness of the CSBB method. Experimental results demonstrate that CSBB effectively stabilizes the step size, preventing it from approaching zero or becoming negative. Moreover, the CSBB method enhances the convergence speed and classification accuracy compared to the traditional BB step size, making it a promising alternative for improving optimization algorithms in machine learning applications.**

***Index Terms*—Barzilai-Borwein (BB) step size, Cumulative Stabilized Barzilai -Borwein (CSBB) Method, optimization problem, adaptive learning rate, machine learning**

I. Introduction

MACHINE learning has rapidly advanced, drawing significant attention from both researchers and industry professionals. This field has emerged as a major focus of research, greatly influencing areas like speech recognition, machine translation, recommendation systems, and image processing. Optimization is essential in machine learning, as numerous tasks in this field can be reframed as optimization problems. At its core, the majority of machine learning algorithms entail constructing an optimization model and deriving the objective function's parameters based on the given data.

To further enhance machine learning methods, numerous effective optimization techniques have been developed to boost their performance and efficiency. From the viewpoint of how gradient information is applied in optimization, commonly used methods fall into three categories: first-order methods, with stochastic gradient approaches being the most notable; high-order methods, exemplified by Newton's method; and heuristic, derivative-free methods, such as the coordinate descent technique discussed in [1].

More recently, the step size selection from the traditional full-gradient method, known as limited memory steepest descent, was modified by Franchini et al. [2]. This modification made the method compatible with stochastic gradient algorithms.

Reference [3] introduced the Barzilai-Borwein (BB) method. This method has since become widely recognized as an efficient approach for tackling large-scale unconstrained optimization problems with moderate precision. Moreover, it can be adapted to address a broad range of constrained optimization problems. Recently, numerous researchers have focused on improving the step size selection in existing algorithms, yielding promising outcomes. For instance, an innovative mini-batch algorithm named mS2GD-BB was presented in [4]. This algorithm integrates the BB method's self-adjusting step size into the mini-batch semi-stochastic gradient descent (mS2GD) framework, which was originally developed in [5]. Similarly, the algorithm SGD-BB was developed by applying the BB method to automatically determine step sizes in Stochastic Gradient Descent (SGD). A similar approach was used to develop SVRG-BB, which applies the BB method to the Stochastic Variance Reduced Gradient (SVRG) variant, as shown in [6]. For a more detailed discussion of the BB method and its applications, readers are referred to [7] and the relevant references. Additionally, a diagonal BB step size (DBB) was proposed in [8] for the variable metric proximal gradient (VM-PG) algorithm. Recently, the DBB approach was implemented in [9] in a mini-batch proximal stochastic recursive gradient algorithm. The convergence of the method was analyzed under various conditions.

From the above-mentioned studies, it is evident that the BB step size plays a crucial role in optimization algorithms. However, due to the fractional structure of the BB step size, situations where the denominator approaches zero are inevitable. Therefore, some scholars have considered improving the BB step size to address this issue. For example, the Stabilized Barzilai-Borwein (SBB) step size was proposed in [10]. This variant adds a positive term to the denominator's absolute value, helping to mitigate the instability issues inherent in the original BB step size. A

Manuscript received February 15, 2025; revised May 13, 2025.

This work was supported in part by Scientific Research Project of the Hunan Provincial Department of Education in 2024 under Grant 24B0819.

Weijuan Shi is a lecturer at College of Mathematics and Finance, Hunan University of Humanities, Science and Technology, Loudi, 417000, China. (e-mail: shiweijuan2007@sina.com).

Adibah Shuib is an Associate Professor at School of Mathematical Sciences. College of Computing, Informatics and Media (KPPIM), Universiti Teknologi MARA (UiTM), 40450 Shah Alam, Selangor, Malaysia. (corresponding author to provide phone: +60192622348; fax: +603-55435501; e-mail: adibah253@uitm.edu.my).

Zuraida Alwadood is a senior lecturer at School of Mathematical Sciences. College of Computing, Informatics and Media (KPPIM), Universiti Teknologi MARA (UiTM), 40450 Shah Alam, Selangor, Malaysia. (e-mail: zuraida794@uitm.edu.my).

stabilized BB method was presented in [11], which limits the distance between successive iterations. This effectively decreases the number of required BB iterations. Reference [12] proposed an improved BB method, called Random Barzilai-Borwein (RBB). The concept of online step size (OSS) was introduced in [13]. Reference [14] resorted the BB technique with a DBB step size to implemented mS2GD algorithm. Reference [15] developed a new adaptive strategy, Positive Defined Stabilized Barzilai-Borwein (PDSBB), which adjusts the BB method to compute step sizes dynamically. The PDSBB method was combined with SGD by the researchers in [16], forming a new algorithm called SGD-PDSBB. The effectiveness of the algorithm was verified. Algorithms such as loopless SVRG (L-SVRG) in [17] improve efficiency by eliminating external loops, thereby reducing computational complexity. Inspired by these improvements to the BB step size, a new variant of this approach is presented in this study.

To further illustrate the current research landscape and practical value of the Barzilai-Borwein (BB) step size method in the field of machine learning, this study analyzes 586 publications from the Web of Science database published between 2016 and 2025. These papers were selected based on the topics "Barzilai-Borwein step size" or "Barzilai-Borwein method" combined with "machine learning". Using CiteSpace, we conducted a comprehensive clustering and visualization analysis from multiple perspectives, including subject categories, keyword clustering, and burst detection of keywords.

Figure 1 presents a circular view of the subject categories involved in BB-related research from 2016 to 2025. It shows that the BB method is widely used not only in traditional fields like Applied Mathematics (#1), Engineering, Electrical & Electronic (#2), and Mechanics (#3), but also in many interdisciplinary areas. These include Radiology and Medical Imaging (#7), Surgery (#4), Energy & Fuels (#5), and Transportation Science & Technology (#0). The broad distribution of these categories suggests that the BB method is becoming a practical and adaptable tool for solving optimization problems in a wide range of fields. Its growing presence across disciplines highlights both its academic value and real-world potential.

Figure 2 shows how the research focus on the BB method has evolved over time. From 2016 to 2018, most studies centered on classical optimization techniques such as conjugate gradient methods, line search strategies, and unconstrained optimization. Later, new topics like variance reduction, image reconstruction, and nonmonotone line search started to appear, reflecting a broader range of applications. Around 2021, "machine learning" became a core keyword, indicating that the BB step size method is gaining importance in the AI field. The timezone view clearly reveals this shift from theoretical optimization to more practical, data-driven research.

Figure 3 shows the top-ranked keyword bursts in BB-related research from 2016 to 2025. "Machine learning" has the highest burst strength (7.1) among all the keywords. This suggests a growing research interest in applying BB methods to machine learning problems. This suggests that combining step size strategies like BB with machine learning has become a key area of focus in recent years. In addition, keywords such as "stepsize" and "adaptive step size" also show strong bursts starting in 2021 and 2023, respectively. These highlight a growing interest in how step sizes are chosen and adjusted during optimization. This trend supports the motivation behind the CSBB method proposed in this paper. The method uses historical information to stabilize the step size. This makes it especially useful for deep learning, where optimization problems often show high variability and non-convex patterns.

Fig. 3. Top-Ranked Keyword Bursts by Strength in BB Method Research (2016 - 2025)

In summary, the literature visualization analysis from multiple perspectives highlights a clear upward trend in the use of the Barzilai-Borwein step size method and its variants in recent machine learning research. The method has shown promising performance and broad application potential in practical areas such as neural network training, image reconstruction, and compressed sensing. Therefore, the CSBB method proposed in this paper is not only theoretically innovative but also holds strong potential for industrial application.

The convergence analysis of these algorithms is not detailed in this paper. This paper focuses only on presenting the newly proposed approach for improving the BB step size, the specific formula of CSBB, and demonstrates the performance of the resulting step sizes and $\left\|g_k\left(x\right)\right\|$.

II. Methodology

To start, it is essential to identify the limitations of the BB step size. We will first examine and analyze its shortcomings. Through theoretical analysis and existing literature review, we pinpoint specific issues such as denominator approaches zero or even negative.

Secondly, propose the CSBB step size. Based on the identified limitations, we propose an improved step size, referred to as the CSBB step size. The CSBB is designed to address the issues identified in the BB step size by introducing a parameter to automatically adjust the step size and stabilize it at an appropriate value. The formulation of the CSBB step size is described in detail, including the theoretical reasoning behind the modifications.

Thirdly, numerical experiments were designed. To confirm the effectiveness of the CSBB step size, two examples were carried out. The examples are common test functions used in optimization algorithms.

Lastly, the evaluation and comparison were finished. We compare the performance of the BB step size, PDSBB step size and CSBB step size and the norm gradient of each step size. To emphasize the advantages of the CSBB step size, this study evaluates the results from the numerical experiments in comparison with the original BB step size. The research framework of this study is illustrated in figure 4.

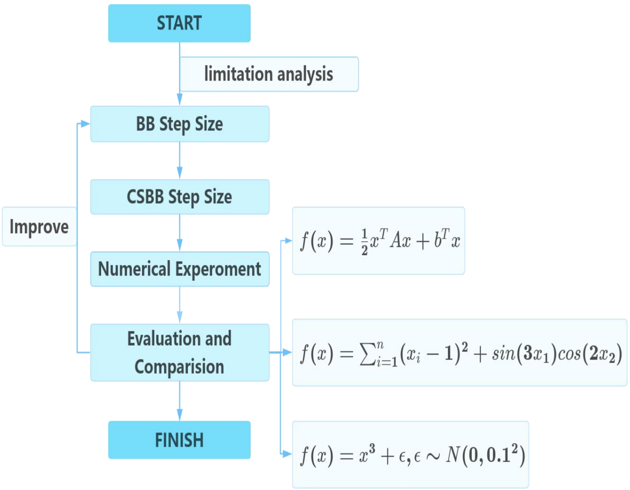

Fig. 4. The Research Framework

III. Cumulative Stabilized Barzilai -Borwein (CSBB) Method

In optimization algorithms, particularly gradient descent, the terms "step size" and "learning rate" are frequently synonymous, both indicating the magnitude of parameter adjustments made during each iteration of the optimization process. A significant challenge in stochastic optimization involves choosing a suitable step size, as it can affect the algorithm's convergence ability. A step size that is excessively large may prevent convergence, while one that is too small can lead to sluggish progress. Thus, finding the right step size is essential for optimal algorithm performance.

The BB method is a gradient-based approach that draws inspiration from Newton’s method, featuring adjusted step sizes and being used alongside a non-monotone line search. A defining feature of the BB method and its variants is their quasi-Newton characteristics. Typically, the method employs the full gradient of the objective function to maintain these characteristics. In the steepest descent method, the BB technique operates as a two-point step size by utilizing an approximation of the secant equation.

This method requires only the storage of additional iterations and gradients, which allows it to achieve lower computational costs while significantly enhancing the performance of classical steepest descent methods and other standard gradient techniques.

Now, consider the following unconstrained optimization problem that needs to be addressed:

$$\min f(w) \tag{1}$$

where $f(w)$ is differentiable. For the optimization problem (1), the iterative formula employed in the quasi-Newton method is given by:

$$w_{t+1} = w_t - B_t^{-1}\nabla f(w_t) \tag{2}$$

The Hessian matrix ($\eta_t > 0$) is approximated through the use of $B_t = \frac{1}{\eta_t} I$, and this approximation is substituted into the secant equation $B_t s_t = y_t$, where $s_t = w_t - w_{t-1}$, $y_t = \nabla f(w_t) - \nabla f(w_{t-1}), t > 1$. By resolving the residual of secant equation, i.e.,

$$\min \left\| \left(\frac{1}{\eta_t s_t} - y_t \right) \right\|^2 \tag{3}$$

the BB step size can be obtained, as

$$\eta_t^{BB1} = \frac{\|s_t\|^2}{s_t^T y_t} \tag{4}$$

Another form of BB step size is given by

$$\eta_t^{BB2} = \frac{s_t^T y_t}{\|y_t\|^2} \tag{5}$$

which is obtained by solving the following:

$$\min \|s_t - \eta_t y_t\|^2 \tag{6}$$

In general, Equation (4) tends to yield better numerical performance than Equation (5) in practical applications. Therefore, this study will primarily focus on Equation (4) as the main starting point. The step size variant will be derived from Equation (3).

Building on the limitations of the BB method previously outlined and the need for stabilization, a new dynamic adaptive step size named Cumulative Stabilized Barzilai-Borwein (CSBB) has been developed. This method modifies the BB technique to compute step sizes automatically. The primary goal of CSBB is to address the issue that arises as the BB step size's denominator approaches zero. When the computed denominator falls below a specified positive threshold, the CSBB method automatically incorporates historical step sizes. The detailed description of this approach is given by:

first, the step size η_t is computed using equation (4). Secondly, when the denominator is close to 0, $s_t^T y_t$ will be compared to the given positive parameter ε, if $s_t^T y_t \le \varepsilon$, set $\eta_t = \frac{1}{t} \cdot \eta_{t-2} + \eta_{t-1}$. That is:

$$\eta_t = \begin{cases} \dfrac{\|s_t\|^2}{s_t^T y_t}, & s_t^T y_t > \varepsilon \\ \dfrac{1}{t} \cdot \eta_{t-2} + \eta_{t-1}, & s_t^T y_t \le \varepsilon \end{cases} \tag{6}$$

The similarity between the CSBB method and the BB step size is noteworthy. When the denominator is greater than a specified parameter ε , the step size produced by the CSBB method corresponds to the BB step size. If, however, the denominator $s_t^T y_t$ falls below a certain threshold ε , the CSBB method calculates the step size as a combination of the two previous step sizes. Specifically, it is determined by adding the step size from the previous iteration to a weighted component of the step size from the iteration prior to that. This can be expressed as the previous step size plus the step size from two iterations earlier, scaled by $\frac{1}{t}$, where t is the current iteration number. This method allows for automatic adjustment of the step size, stabilizing it at an appropriate value. Additionally, it effectively prevents situations where the denominator of the step size approaches zero or even becomes negative.

The CSBB method introduced in this paper serves as foundational work for machine learning research and will be integrated with various stochastic optimization algorithms. This includes various methods such as SGD proposed in [18]; the SVRG method introduced in [19]; SAGA, an innovative incremental gradient approach introduced in [20]; the mS2GD approach developed in [5]; the Stochastic Average Gradient (SAG) method created in [21]; the Stochastic Path-Integrated Differential Estimator (SPIDER) method presented in [22]; the Accelerated Mini-batch Prox-SVRG (Acc-Prox-SVRG) algorithm introduced in [23]; and the Stochastic Recursive Gradient Algorithm (SARAH) proposed in [24], among others, all aimed at improving the optimization process and computational efficiency of these algorithms.

IV. Numerical Example and Discussion

The numerical performance of the CSBB method is illustrated in this section through two numerical experiments. All coding was carried out in MATLAB 2022 and executed on a personal computer equipped with a 12th Gen Intel(R) Core (TM) i5-12500H processor, operating at 2500 MHz with 12 cores, and running on the Windows 11 operating system. The tolerance is set to $tol = 10^{-8}$, and the termination criterion is $\|g_k(x)\| < tol$. The maximum number of iterations is configured to 40. This paper mainly presents the results of two examples.

In the two examples in this paper, the CSBB method proposed is consistently represented by a solid line with triangles, the BB method is depicted using a solid line with circles, while the PDSBB method is illustrated with a solid line featuring crosses. Example 1 is similar to the one presented in reference [1], and Example 2 is a common test function used in optimization algorithms or numerical analysis to assess performance.

A. *Example 1*

In this example, we minimize the function

$$f(x) = \tfrac{1}{2} x^T A x - b^T x$$

where $A = \begin{pmatrix} 200 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 10 \end{pmatrix}$, $b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$.

Figure 5 displays the changes in step sizes over the iterations, along with the corresponding gradient sequence $\{\| g_k \|\}$ (after taking log10) for both the BB, PDSBB and CSBB methods on the given function in this example. The initial step size for all three methods is set to 1.

Figure 5 illustrates that, in Example 1, the CSBB step size exhibits a divergent trend from that of BB and PDSBB after approximately 20 iterations. Upon completion of the 35th iteration, the results for the BB step size and the CSBB step size are identical. In overall, the differences in step sizes among the three methods are not significant. This is due to the fact that the step sizes of BB remain relatively stable in this example, which means that the step size of CSBB does not undergo significant revisions. This ensures that the step size maintains a decreasing property, which is consistent with the strategy of using a decreasing step size in machine learning optimization algorithms. However, in terms of the gradient norm, none of the three step sizes are very stable and all exhibit some degree of oscillation. Comparatively, the BB step size is the most stable one.

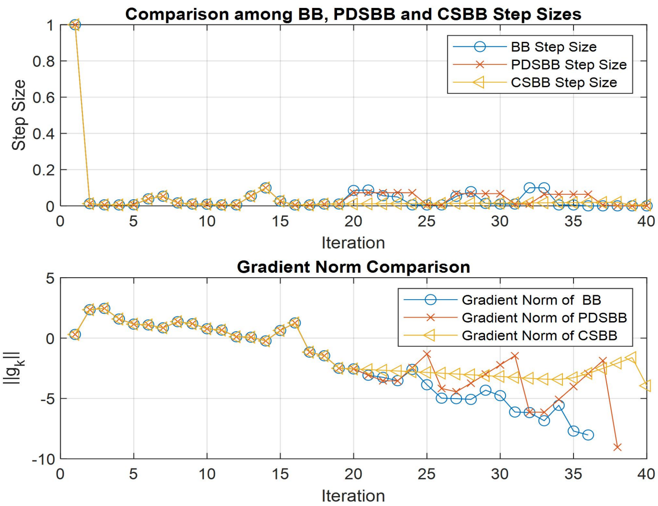

Fig. 5. Step Size and $\| g_k \|$ Results of the BB, PDSBB and CSBB Methods on Example 1

To facilitate the presentation of results, we present the results of the gradient norm in Table I. The values of the Norm Gradient (NG) for three different step sizes in this example are presented in the table.

TABLE I
RESULTS OF $\| g_k \|$ FOR THE BB, PDSBB AND CSBB METHODS

Iteration	NG_BB	NG_PDSBB	NG_CSBB
1	2.00000E+00	2.00000E+00	2.00000E+00
2	2.23258E+02	2.23258E+02	2.23258E+02
3	2.84675E+02	2.84675E+02	2.84675E+02
4	3.79879E+01	3.79879E+01	3.79879E+01
5	1.42709E+01	1.42709E+01	1.42709E+01
6	1.22516E+01	1.22516E+01	1.22516E+01
7	7.04790E+00	7.04790E+00	7.04790E+00
8	2.28707E+01	2.28707E+01	2.28707E+01
⋮	⋮	⋮	⋮
31	7.51687E-07	3.43216E-02	5.58080E-04
32	6.76314E-07	7.30671E-07	4.70026E-04
33	1.43188E-07	7.22781E-07	3.99282E-04
34	2.70355E-06	8.30614E-06	3.73174E-04
35	2.00160E-08	9.74384E-05	5.24164E-04
36	9.47780E-09	1.14305E-03	1.17449E-03
37	0.00000E+00	1.34092E-02	3.07445E-03
38	0.00000E+00	9.10932E-10	8.46312E-03
39	0.00000E+00	0.00000E+00	2.41454E-02
40	0.00000E+00	0.00000E+00	1.12707E-04

B. *Example 2 Consider the function*

$$f(x) = \sum_{i=1}^{n} (x_i - 1)^2 + sin(3x_1) cos(2x_2)$$

where n is taken to be 100, 400, 800, and 1600, respectively.

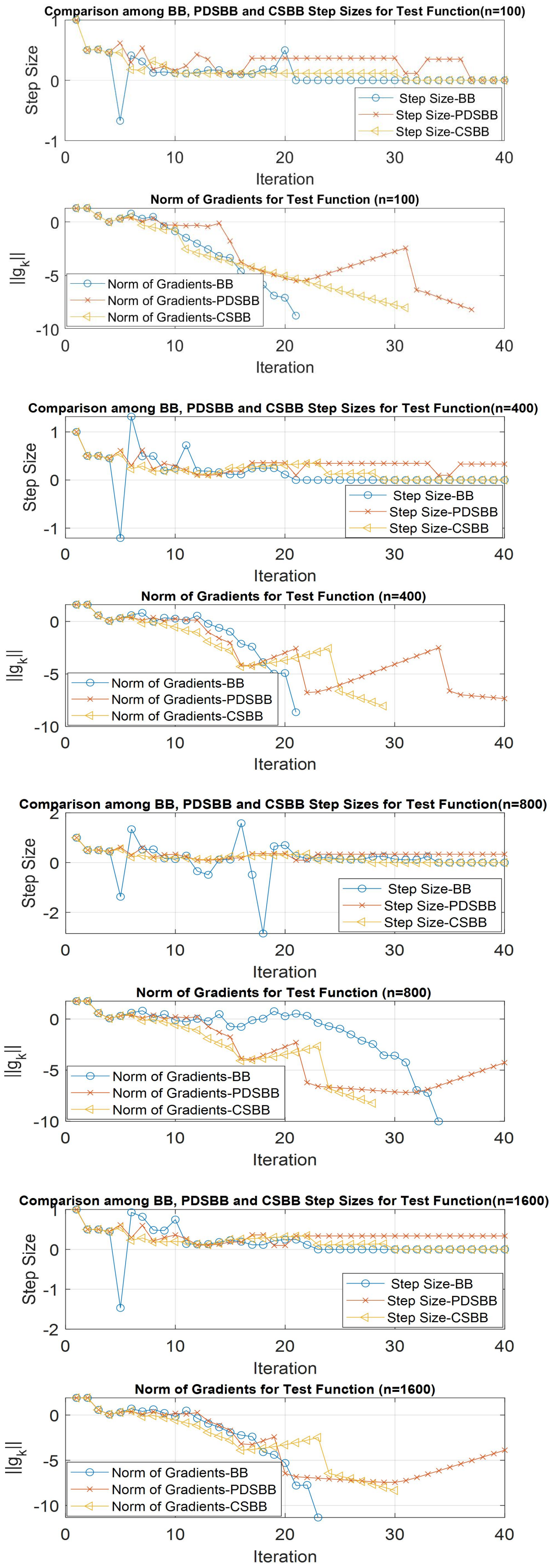

Fig. 6. Step Size and $\| g_k \|$ Results of the BB, PDSBB and CSBB Methods on Example 2 with $n=100$ (Fig. 6 a)), $n=400$ (Fig. 6 b)), $n=800$ (Fig. 6 c)), and $n=1600$ (Fig. 6 d)), respectively.

Figure 6 illustrates the variations of step sizes with the iteration and the resulting gradient sequence $\{\| g_k \|\}$ (after taking log10) for both the BB, PDSBB and CSBB methods on the provided example 2.

From Figures 6 a), b), and d), it is evident that when the step size approaches zero (i.e., less than a positive threshold set $\varepsilon = 10^{-6}$ in this paper) or becomes negative, the CSBB step size effectively prevents these issues and maintains a more appropriate value. This capability is particularly advantageous when employing the BB step size in conjunction with first-order stochastic optimization algorithms, as it mitigates the risk of algorithm failure. This principle underpins the design of the CSBB step size.

In comparison, the convergence speed of the CSBB method is only slightly slower than that of the BB method; the BB method achieves convergence tolerance in approximately 20 steps, while the CSBB method requires about 30 steps, as illustrated in the second subplots of Figures 6 a), b), and d). The CSBB method successfully avoids negative step sizes while minimally impacting the convergence speed, indicating that it stabilizes the step size with negligible cost.

In Figure 6 c), it is observed that as the parameter n increases, the BB step size exhibits substantial fluctuations, resulting in multiple negative values. In contrast, both the PDSBB and CSBB methods effectively stabilize the step size.

Regarding $\| g_k \|$, the CSBB method consistently reaches the stopping criterion before the PDSBB method. When $n = 800$, the CSBB method reaches the stopping criterion in approximately 28 iterations, compared to about 34 iterations for the BB method. This demonstrates that the CSBB method not only prevents the denominator of the step size from nearing zero or becoming negative but also enhances convergence speed.

Overall, this example illustrates that the CSBB method stabilizes the step size with minimal impact on convergence speed in some instances, while in others, it improves both stability and convergence efficiency.

TABLE II
RESULTS OF $\| g_k \|$ FOR THE BB, PDSBB AND CSBB METHODS WITH $n = 100$

Iteration	NG_BB	NG_PDSBB	NG_CSBB
1	1.99249E+01	1.99249E+01	1.99249E+01
2	1.99857E+01	1.99857E+01	1.99857E+01
3	3.73893E+00	3.73893E+00	3.73893E+00
4	1.01578E+00	1.01578E+00	1.01578E+00
5	2.09272E+00	2.09272E+00	2.09272E+00
6	6.18328E+00	2.52978E+00	2.52978E+00
7	2.05955E+00	1.14855E+00	1.14855E+00
8	3.07247E+00	2.24186E+00	2.24186E+00
9	4.05345E-01	5.25731E-01	5.25731E-01
10	1.37204E-01	5.22766E-01	5.22766E-01
⋮	⋮	⋮	⋮
32	0.00000E+00	4.50201E-07	4.50201E-07
33	0.00000E+00	2.26664E-07	2.26664E-07
34	0.00000E+00	9.31507E-08	9.31507E-08
35	0.00000E+00	3.82817E-08	3.82817E-08
36	0.00000E+00	1.57328E-08	1.57328E-08
37	0.00000E+00	6.46985E-09	6.46985E-09
38	0.00000E+00	0.00000E+00	0.00000E+00
39	0.00000E+00	0.00000E+00	0.00000E+00
40	0.00000E+00	0.00000E+00	0.00000E+00

TABLE III
RESULTS OF $\|g_k\|$ FOR THE BB, PDSBB AND CSBB METHODS WITH $n = 400$

Iteration	NG_BB	NG_PDSBB	NG_CSBB
1	3.99625E+01	3.99625E+01	3.99625E+01
2	3.99929E+01	3.99929E+01	3.99929E+01
3	3.73476E+00	3.73476E+00	3.73476E+00
4	1.13699E+00	1.13699E+00	1.13699E+00
5	2.04568E+00	2.04568E+00	2.04568E+00
6	4.00220E+00	2.47395E+00	2.96584E+00
7	6.70169E+00	1.28185E+00	7.12842E-01
8	9.50151E-01	2.36812E+00	8.86258E-01
9	2.25306E+00	1.12687E+00	5.26391E-01
⋮	⋮	⋮	⋮
34	0.00000E+00	3.18473E-03	0.00000E+00
35	0.00000E+00	2.44682E-07	0.00000E+00
36	0.00000E+00	1.05561E-07	0.00000E+00
37	0.00000E+00	8.54954E-08	0.00000E+00
38	0.00000E+00	6.92442E-08	0.00000E+00
39	0.00000E+00	5.60821E-08	0.00000E+00
40	0.00000E+00	4.54221E-08	0.00000E+00

TABLE IV
RESULTS OF $\|g_k\|$ FOR THE BB, PDSBB AND CSBB METHODS WITH $n = 800$

Iteration	NG_BB	NG_PDSBB	NG_CSBB
1	5.65420E+01	5.65420E+01	5.65420E+01
2	5.65635E+01	5.65635E+01	5.65635E+01
3	3.73407E+00	3.73407E+00	3.73407E+00
4	1.15676E+00	1.15676E+00	1.15676E+00
5	2.03061E+00	2.03061E+00	2.03061E+00
6	4.11054E+00	2.48399E+00	2.97717E+00
7	6.50970E+00	1.26848E+00	7.15265E-01
8	1.36893E+00	2.39523E+00	8.75926E-01
⋮	⋮	⋮	⋮
31	5.72383E-05	6.46326E-08	0.00000E+00
32	1.10648E-07	7.17122E-08	0.00000E+00
33	5.93308E-08	1.27640E-07	0.00000E+00
34	9.54804E-11	2.89516E-07	0.00000E+00
35	0.00000E+00	6.86805E-07	0.00000E+00
36	0.00000E+00	1.63885E-06	0.00000E+00
37	0.00000E+00	3.91345E-06	0.00000E+00
38	0.00000E+00	9.34583E-06	0.00000E+00
39	0.00000E+00	2.23193E-05	0.00000E+00
40	0.00000E+00	5.33022E-05	0.00000E+00

TABLE V
RESULTS OF $\|g_k\|$ FOR THE BB, PDSBB AND CSBB METHODS WITH $n = 1600$

Iteration	NG_BB	NG_PDSBB	NG_CSBB
1	7.99812E+01	7.99812E+01	7.99812E+01
2	7.99964E+01	7.99964E+01	7.99964E+01
3	3.73372E+00	3.73372E+00	3.73372E+00
4	1.16659E+00	1.16659E+00	1.16659E+00
5	2.02232E+00	2.02232E+00	2.02232E+00
6	5.21453E+00	2.49094E+00	2.98440E+00
7	2.52022E+00	1.25833E+00	7.15555E-01
8	4.31962E+00	2.40471E+00	8.67379E-01
9	1.68441E+00	9.83728E-01	5.19124E-01
⋮	⋮	⋮	⋮
30	0.00000E+00	3.66928E-08	4.57161E-09
31	0.00000E+00	5.75012E-08	0.00000E+00
32	0.00000E+00	1.25690E-07	0.00000E+00
33	0.00000E+00	2.96479E-07	0.00000E+00
34	0.00000E+00	7.06677E-07	0.00000E+00
35	0.00000E+00	1.68659E-06	0.00000E+00
36	0.00000E+00	4.02595E-06	0.00000E+00
37	0.00000E+00	9.61026E-06	0.00000E+00
38	0.00000E+00	2.29405E-05	0.00000E+00
39	0.00000E+00	5.47610E-05	0.00000E+00
40	0.00000E+00	1.30719E-04	0.00000E+00

To facilitate the presentation of results, we present the results of the gradient norm in Table II, Table III, Table IV, and Table V. These tables show the values of the norm gradient for three different step sizes in this example, with $n = 100$, $n = 400$, $n = 800$, and $n = 1600$, respectively.

C. Example 3 Comparison of SGD-BB and SGD-CSBB in Neural Network Training

This experiment aims to compare the performance of SGD-BB and SGD-CSBB in neural network training, focusing on convergence speed, stability, and sensitivity to the initial learning rate.

In this experiment, we use PyTorch to train a simple neural network with a 1-10-1 architecture for a regression task: fitting the nonlinear function $y = x^3$ with Gaussian noise (mean of 0 and standard deviation of 0.1). The model architecture includes a single hidden layer with 10 neurons, using ReLU activation and mean squared error (MSE) as the loss function. We compare two optimization methods: SGD-BB, which employs the Barzilai-Borwein (BB) step size, and SGD-CSBB, which adopts the Cumulative Stabilized BB (CSBB) step size. The training is conducted for 100 epochs under three different initial learning rates (0.01, 0.05, and 0.1) in a GPU-supported environment.

Figure 7 illustrates how the loss function changes during training for different learning rates (lr = 0.01, 0.05, 0.1). When the learning rate is set to 0.01, the SGD-BB method (dashed lines) shows severe fluctuations and even diverges during training. In contrast, the SGD-CSBB method (solid lines) steadily converges to a lower loss value. This difference suggests that the CSBB step size strategy provides better stability when the learning rate is low.

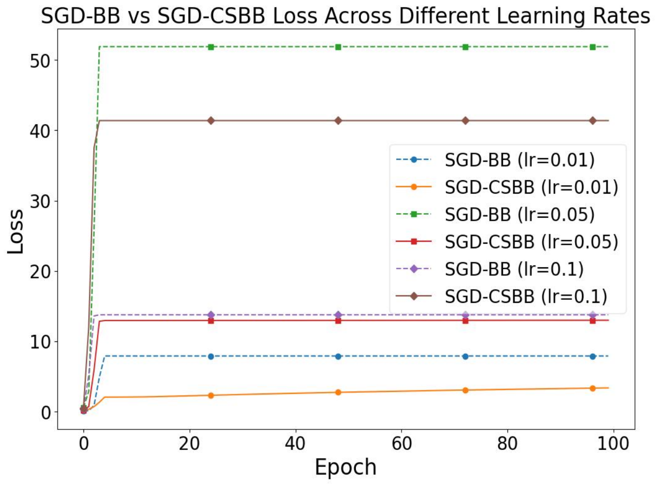

Fig. 7. Comparison of Loss Between SGD-BB and SGD-CSBB Under Varying Initial Learning Rates.

The results indicate that CSBB achieves a more stable convergence curve across all initial learning rates, effectively mitigating the oscillations observed in BB when using a higher learning rate (e.g., 0.1). Additionally, CSBB exhibits improved stability in the early training stages, leading to a lower initial loss. These results clearly indicate that the CSBB method offers superior convergence behavior across different learning rates, making it a robust choice for optimizing the model.

As shown in Table VI, the computational time of CSBB is comparable to BB, with an average increase of approximately 0.005s, which is a negligible trade-off for enhanced stability.

TABLE VI
RESULTS OF COMPUTATIONAL TIME FOR THE BB AND CSBB METHODS

Initial Learning Rate	BB Time	CSBB Time
0.01	0.1159	0.1216
0.05	0.1214	0.1227
0.1	0.1144	0.1219

Overall, the CSBB method proposed in this paper demonstrates a robust step size adaptation strategy, offering smooth convergence, which is not sensitivity to the initial learning rate, and it also can reduce computational overhead, making it a preferable choice for optimization tasks requiring in neural network training.

D. Example 4 Performance Comparison of BB and CSBB Step Size Strategies in Binary Classification

In this experiment, we will compare two step size strategies—BB and CSBB—in a binary classification problem. Using the make_moons function, we will generate a dataset of two interleaving half-moon shapes with added noise to reflect real-world complexities. A simple feedforward neural network will be built for this classification task. We will train the network using Stochastic Gradient Descent (SGD), applying both BB and adaptive CSBB step size strategies. During training, we will track the training loss, and classification accuracy at each epoch. By plotting the changes in loss convergence, and accuracy—using different line styles and markers—we aim to clearly assess the performance differences between the BB and CSBB strategies.

The experimental results show that SGD-CSBB performs better in terms of convergence and classification accuracy during training. In the loss convergence curve (Figure 8), SGD-CSBB (solid line) decreases faster in the early stages and eventually converges to a lower loss than SGD-BB (dashed line). This suggests that the CSBB step size strategy adapts more effectively during optimization, leading to more stable training and a lower final loss. In contrast, SGD-BB adjusts the step size less efficiently, resulting in slower convergence and a higher final loss.

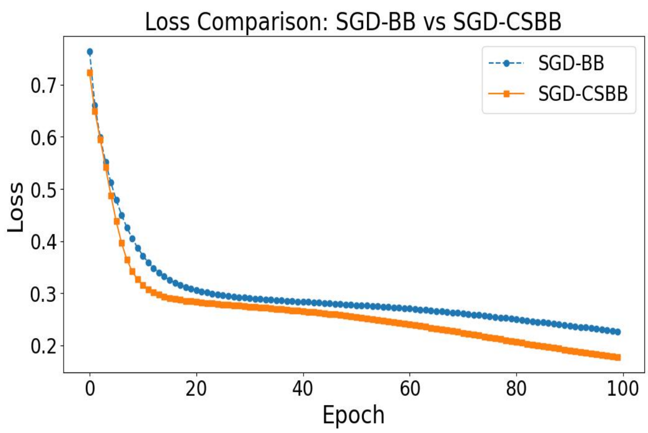

Fig. 8. Loss Comparison Between SGD-BB and SGD-CSBB

In the accuracy comparison (Figure 9), SGD-CSBB improves accuracy more rapidly at the beginning of training and reaches convergence within fewer epochs. It ultimately achieves an accuracy of over 92%, while SGD-BB stabilizes at around 90%. This indicates that CSBB can find the optimal step size more quickly, improving both convergence speed and final performance.

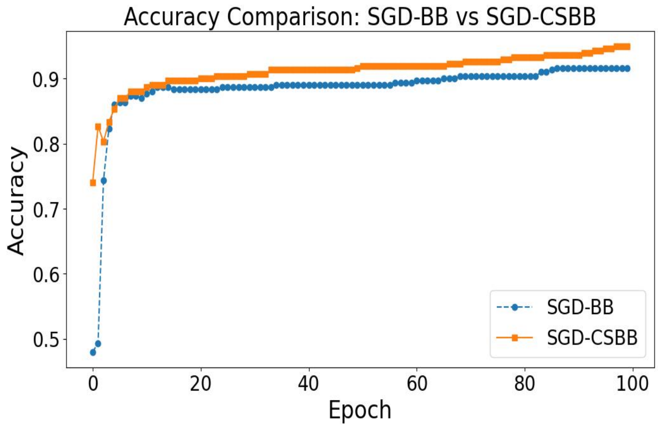

Fig. 9. Accuracy Comparison Between SGD-BB and SGD-CSBB

Overall, the results demonstrate that compared to the traditional BB step size, CSBB achieves better optimization in binary classification tasks by converging faster and yielding higher accuracy.

V. INDUSTRY APPLICATIONS

The BB method is widely recognized for its simplicity and effectiveness in large-scale optimization tasks. In this study, we searched the Web of Science database using the topics “Barzilai-Borwein step size” or “Barzilai-Borwein method”, retrieving 8,744 relevant publications. We then used the CiteSpace software to conduct a keyword clustering analysis based on publications from the past decade (2016 – 2025). Figure 10 presents the cluster view of keywords, where each color denotes a distinct research theme.

As shown in Figure 10, the BB method is not only strongly associated with classical optimization topics such as “global optimization” and “model”, but also shows significant connections with emerging application areas like “machine learning” and “biomarkers”. In particular, Cluster #10 (“barzilai-borwein method”) demonstrates strong linkages with Cluster #5 (“global optimization”) and Cluster #6 (“machine learning”), indicating the widespread adoption of BB-based techniques in intelligent algorithms and practical problem-solving.

The color gradient of the nodes, representing the average publication year, suggests that research on BB methods has remained active in recent years. This illustrates the method’s ongoing relevance and promising future in both numerical optimization and machine learning.

Several interdisciplinary clusters are also observed, such as “oxidative stress”, “growth factor”, and “adsorption”, indicating the expansion of BB-related methods into fields including biomedical science, environmental science, and materials science.

In summary, this analysis reveals that the BB method has evolved from a technique primarily used in mathematical optimization to a broadly applicable tool across various scientific and engineering domains. Its flexibility and efficiency support its use in solving real-world problems, thereby motivating the development of our proposed CSBB method, which aims to enhance the stability of step size selection in deep learning and other complex tasks.

Despite its advantages, the performance of the BB method can degrade when the denominator in the step size formula approaches zero or becomes negative, leading to instability and unreliable convergence. To address these limitations, the proposed CSBB method introduces a correction mechanism that averages the BB step sizes from the two most recent iterations when such anomalies occur.

As a result, the CSBB method proves effective in training machine learning models where gradient-based optimization is crucial. For example, it can be integrated into stochastic gradient descent (SGD) to dynamically adjust step sizes, making it suitable for neural network training.

Additionally, the CSBB method holds promise in fields such as compressed sensing and medical imaging (e.g., MRI and CT), where high-quality image reconstruction from limited data is critical. It can also improve performance in image restoration, image deblurring, and network optimization by efficiently solving large-scale linear inverse problems.

Additionally, the CSBB method has potential applications in compressed sensing and medical imaging, such as MRI and CT, where reconstructing high-quality images from limited data is essential. It can also enhance optimization performance in tasks like image restoration, image deblurring, and network optimization by solving large-scale linear inverse problems efficiently.

The CSBB method has potential applications in image restoration and network optimization tasks. In imaging and computer vision, CSBB method has potential applications in solving optimization problems for applications such as imaging, recovering sharp images from blurred ones.

VI. Conclusion

This study is motivated by applying the BB step size to optimization in machine learning. The CSBB step size is introduced in this paper, addressing the limitations of the BB step size, along with four examples to illustrate its application. The examples show that the CSBB step size is sometimes very similar to the BB step size. Significant changes occur mainly when the denominator of BB step size gets close to zero or turns negative. The CSBB method stabilizes the step size with a small cost to convergence speed, and in some cases, it even improves the convergence speed while ensuring a stable step size. The step size proposed in this paper also be applied to stochastic optimization algorithms and machine learning models for solving problems. In future research, consideration should be given to the convergence of the new stochastic optimization algorithm.

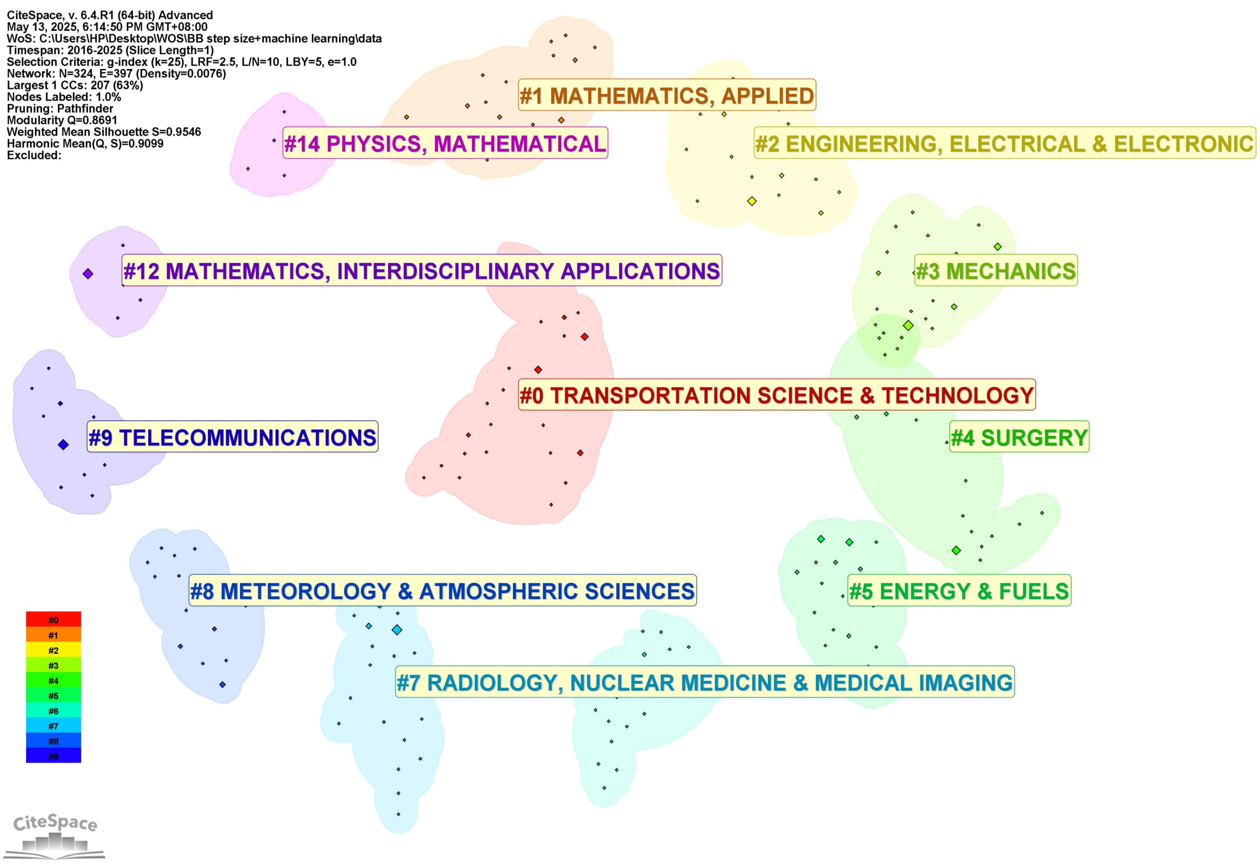

Fig. 1. Circular Visualization of Subject Categories in BB Method Research (2016－2025)

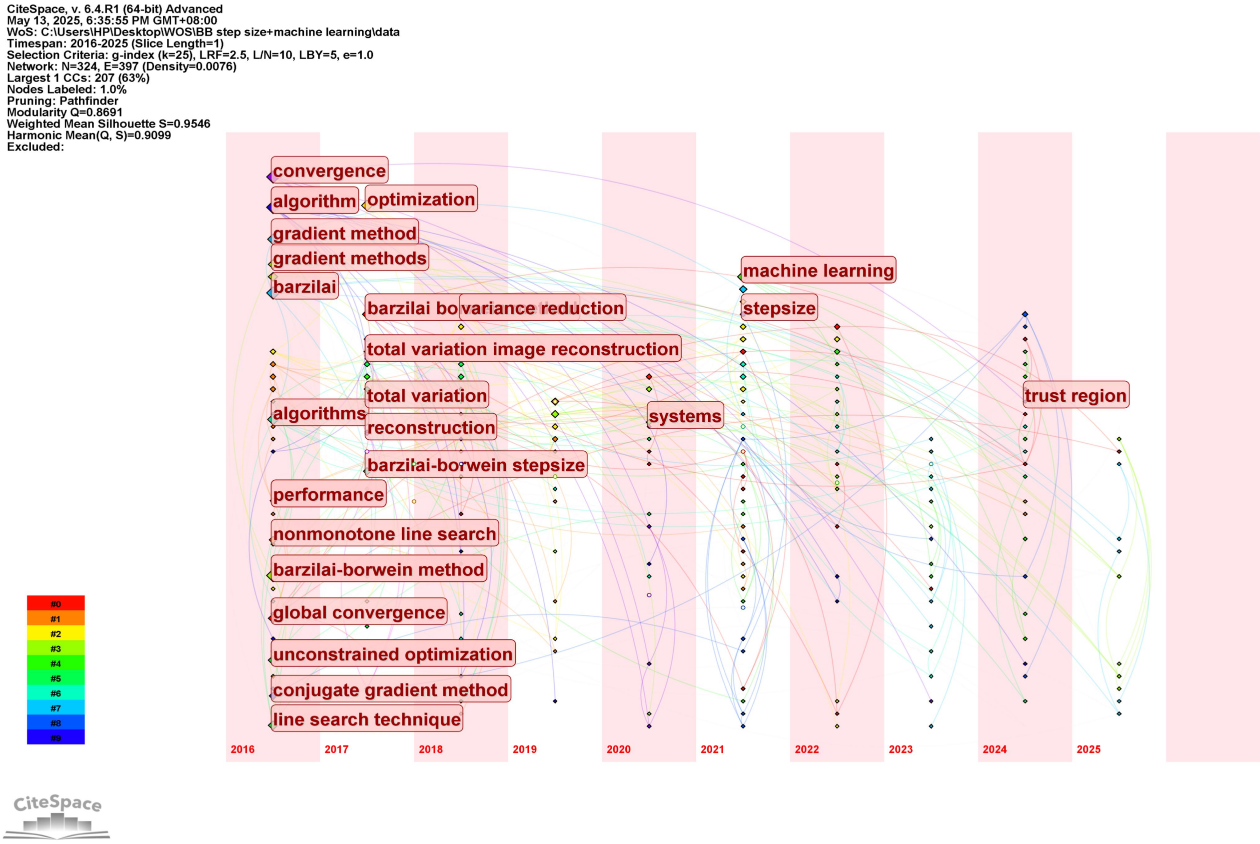

Fig. 2. Temporal Evolution of Keywords in BB Method Research: Timezone View (2016 – 2025)

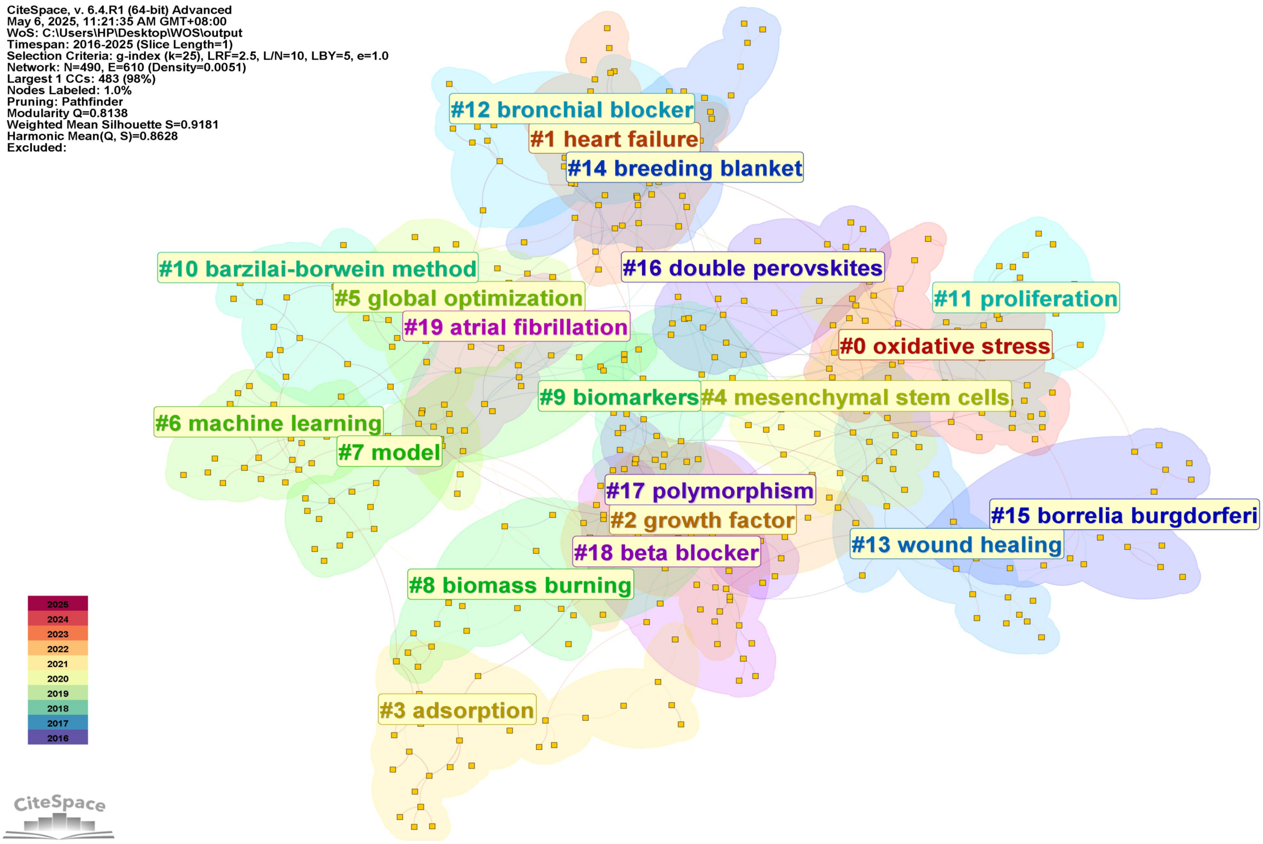

Fig. 10. Keyword Cluster View of BB Method Research (2016–2025)

REFERENCES

[1] S. Sun, Z. Cao, H. Zhu, and J. Zhao, "A Survey of Optimization Methods From a Machine Learning Perspective," *IEEE Trans. Cybernetics*, vol. 50, no. 8, pp. 3668-3681, 2020, doi: 10.1109/TCYB.2019.2950779.

[2] G. Franchini, V. Ruggiero, and L. Zanni, "Ritz-like values in steplength selections for stochastic gradient methods," *Soft Comput.*, vol. 24, pp. 17573–17588, 2020, doi: 10.1007/s00500-020-05219-6.

[3] J. Barzilai and J. Borwein, "Two-Point Step Size Gradient Methods," *IMA J. Numer. Anal.*, vol. 8, pp. 141–148, 1988, doi: 10.1093/imanum/8.1.141.

[4] Z. Yang, C. Wang, Y. Zang, and J. Li, “Mini-batch algorithms with Barzilai–Borwein update step,” *Neurocomputing*, vol. 314, pp. 177–185, 2018.

[5] J. Konečný, J. Liu, P. Richtárik, and M. Takáč, “Mini-Batch Semi-Stochastic Gradient Descent in the Proximal Setting,” *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 2, pp. 242–255, 2016.

[6] C. Tan, S. Ma, Y. H. Dai, and Y. Qian, “Barzilai-Borwein step size for stochastic gradient descent,” *Advances in Neural Information Processing Systems*, pp. 685–693, 2016.

[7] Z. Yang, “On the step size selection in variance-reduced algorithm for nonconvex optimization,” *Expert Syst. Appl.*, p. 114336, 2020.

[8] Y. Park, S. Dhar, S. Boyd, and M. Shah, “Variable metric proximal gradient method with diagonal Barzilai-Borwein stepsize,” in *ICASSP 2020-2020 IEEE Int. Conf. Acoustics, Speech and Signal Process. (ICASSP)*, IEEE, 2020, pp. 3597–3601.

[9] T. Yu, X. W. Liu, Y. H. Dai, and J. Sun, “A variable metric mini-batch proximal stochastic recursive gradient algorithm with diagonal Barzilai-Borwein stepsize,” *arXiv preprint arXiv:201000817*, 2020.

[10] K. Ma, J. Zeng, J. Xiong, Q. Xu, X. Cao, W. Liu, and Y. Yao, “Stochastic Non-convex Ordinal Embedding with Stabilized Barzilai-Borwein step size,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, pp. 3738–3745, 2018.

[11] O. Burdakov, Y. H. Dai, and N. Huang, “Stabilized Barzilai-Borwein method,” *arXiv preprint arXiv:1907.06409*, 2019. [Online]. Available: https://doi.org/10.48550/arXiv.1907.06409.

[12] Z. Yang, C. Wang, Z. Zhang, and J. Li, “Random Barzilai-Borwein step size for mini-batch algorithms,” *Eng. Appl. Artif. Intell.*, vol. 72, pp. 124–135, 2018.

[13] Z. Yang, C. Wang, Z. Zhang, and J. Li, “Mini-batch algorithms with online step size,” *Knowl.-Based Syst.*, vol. 165, pp. 228–240, 2019. [Online]. Available: https://doi.org/10.1016/j.knosys.2018.11.031.

[14] Z. Yang and L. Ma, “Adaptive step size rules for stochastic optimization in large-scale learning,” *Statist. Comput.*, vol. 33, no. 2, 2023. [Online]. Available: https://link.springer.com/article/10.1007/s11222-023-10218-2.

[15] Weijuan Shi, Adibah Shuib, and Zuraida Alwadood, "Stochastic Variance Reduced Gradient Method Embedded with Positive Defined Stabilized Barzilai-Borwein," IAENG International Journal of Applied Mathematics, vol. 53, no.4, pp1682-1687, 2023.

[16] Weijuan Shi, Adibah Shuib, and Zuraida Alwadood, "Stochastic Gradient Descent with Positive Defined Stabilized Barzilai-Borwein method," IAENG International Journal of Computer Science, vol. 52, no. 2, pp333-352, 2025.

[17] D. Kovalev, S. Horváth, and P. Richtárik, “Don’t jump through hoops and remove those loops: SVRG and Katyusha are better without the outer loop,” *Algorithmic Learning Theory*, pp. 451–467, 2020.

[18] H. Robbins and S. Monro, “A Stochastic Approximation Method,” *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, 1951.

[19] R. Johnson and T. Zhang, “Accelerating stochastic gradient descent using predictive variance reduction,” *Advances in Neural Information Processing Systems*, vol. 1, no. 3, pp. 1–9, 2013.

[20] A. Defazio, F. Bach, and S. Lacoste-Julien, “SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives,” *Advances in Neural Information Processing Systems*, vol. 27, pp. 1646–1654, 2014. [Online]. Available: https://doi.org/10.48550/arXiv.1407.0202.

[21] N. L. Roux, M. Schmidt, and F. R. Bach, “A stochastic gradient method with an exponential convergence rate for finite training sets,” *Advances in Neural Information Processing Systems*, pp. 2663–2671, 2013. [Online]. Available: https://doi.org/10.48550/arXiv.1202.6258.

[22] C. Fang, C. J. Li, Z. Lin, and T. Zhang, “SPIDER: Near-Optimal Non-Convex Optimization via Stochastic Path Integrated Differential Estimator,” *arXiv*, abs/1807.01695, 2018.

[23] A. Nitanda, “Stochastic proximal gradient descent with acceleration techniques,” *Neural Information Processing Systems*, vol. 1, pp. 1574–1582, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:5628754.

[24] L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takáč, “SARAH: a novel method for machine learning problems using stochastic recursive gradient,” *Proceedings of the 34th International Conference on Machine Learning*, *Proceedings of Machine Learning Research* (PMLR), vol. 70, pp. 2613–2621, 2017. [Online]. Available: https://doi.org/10.48550/arXiv.1703.00102.

Weijuan Shi is a lecturer at College of Mathematics and Finance, Hunan University of Humanities, Science and Technology since 2014. Her main research interests include stochastic optimization and machine learning.

Adibah Shuib is an Associate Professor at Universiti Teknologi MARA, Malaysia. She received her PhD degree in Mathematics from University of Birmingham, United Kingdom. Her main research fields include Mathematical Programming / Optimization Models, Graph Theory, Transportation, Logistics and Supply Chain.

Zuraida Alwadood received her PhD degree in Information Technology and Quantitative Sciences from the Universiti Teknologi MARA, Malaysia. Her top skills are Mathematical Programming/Modelling.