CONSTRUCTIVE MATHEMATICAL ANALYSIS
8 (2025), No. 2, pp. 50-80
http://dergipark.org.tr/en/pub/cma
ISSN 2651 - 2939

Research Article

Halpern-type relaxed algorithms with alternated and multi-step inertia for split feasibility problems with applications in classification problems

ABDULWAHAB AHMAD, POOM KUMAM*, YEOL JE CHO, AND KANOKWAN SITTHITHAKERNGKIET

ABSTRACT. In this article, we construct two Halpern-type relaxed algorithms with alternated and multi-step inertial extrapolation steps for split feasibility problems in infinite-dimensional Hilbert spaces. The first is the most general inertial method that employs three inertial steps in a single algorithm, one of which is an alternated inertial step, while the others are multi-step inertial steps, representing the recent improvements over the classical inertial step. Besides the inertial steps, the second algorithm uses a three-term conjugate gradient-like direction, which accelerates the sequence of iterates toward a solution of the problem. In proving the convergence of the second algorithm, we dispense with some of the restrictive assumptions in some conjugate gradient-like methods. Both algorithms employ a self-adaptive and monotonic step-length criterion, which does not require a knowledge of the norm of the underlying operator or the use of any line search procedure. Moreover, we formulate and prove some strong convergence theorems for each of the algorithms based on the convergence theorem of an alternated inertial Halpern-type relaxed algorithm with perturbations in real Hilbert spaces. Further, we analyse their applications to classification problems for some real-world datasets based on the extreme learning machine (ELM) with the ℓ_1-regularization approach (that is, the Lasso model) and the $\ell_1 - \ell_2$ hybrid regularization approach. Furthermore, we investigate their performance in solving a constrained minimization problem in infinite-dimensional Hilbert spaces. Finally, the numerical results of all experiments show that our proposed methods are robust, computationally efficient and achieve better generalization performance and stability than some existing algorithms in the literature.

Keywords: Relaxed $\mathcal{CQ}$ method; Alternated inertial method; Multi-step inertial method; Conjugate gradient method, Split feasibility problem; Classification problem.

2020 Mathematics Subject Classification: 47H05, 47J20, 47J25, 47J30, 65K15, 90C25.

1. INTRODUCTION

Throughout this work, let $\mathcal{H}_1$ and $\mathcal{H}_2$ be real Hilbert spaces, $\mathcal{C}$ and $\mathcal{Q}$ denote nonempty closed and convex sets in $\mathcal{H}_1$ and $\mathcal{H}_2$ respectively, and $\mathcal{B} : \mathcal{H}_1 \to \mathcal{H}_2$ be a bounded linear operator. The split feasibility problem, first introduced by Censor and Elfving [10], is the problem of finding a point $x^* \in \mathcal{C}$ such that

$$\mathcal{B}x^* \in \mathcal{Q}. \tag{1.1}$$

Most of the motivations for studying problem (1.1) stem from its usefulness is solving various inverse problems arising from many real-world applications, such as X-ray tomography [41], machine learning [50, 13], image and signal reconstruction and jointly constrained Nash equilibrium [20, 52], to mention but just a few. The primary task in studying problem (1.1) is

Received: 08.10.2024; Accepted: 26.12.2024; Published Online: 05.06.2025
*Corresponding author: Poom Kumam; poom.kum@kmutt.ac.th
DOI: 10.33205/cma.1563173

to develop a robust and efficient numerical algorithm for its solution. Based on the following fixed point problem:

$$x = P_{\mathcal{C}}\big(I - \tau\mathcal{B}^*(I - P_{\mathcal{Q}})\mathcal{B}\big)x \tag{1.2}$$

and the particular case of a Fréchet differentiable real-valued function $g : \mathcal{H}_1 \to \mathbb{R}$ defined by

$$g(x) = \frac{1}{2}||(I - P_{\mathcal{Q}})\mathcal{B}x||^2, \tag{1.3}$$

the iterative algorithm called the $\mathcal{CQ}$ algorithm for solving problem (1.1) was firstly developed by Byrne [7], which is recursively generated for any initial point $x_0 \in \mathcal{H}_1$ by

$$x_{n+1} = P_{\mathcal{C}}\big(x_n - \tau\mathcal{B}^*(I - P_{\mathcal{Q}})\mathcal{B}x_n\big), \quad \forall n \geq 0, \tag{1.4}$$

where $P_{\mathcal{C}} : \mathcal{H}_1 \to \mathcal{C}$ and $P_{\mathcal{Q}} : \mathcal{H}_2 \to \mathcal{Q}$ are the metric (orthogonal) projection operators, I is the identity operator in $\mathcal{H}_1$, $\mathcal{B}^*$ is the adjoint of $\mathcal{B}$ and $\tau \in \left(0, \frac{2}{||\mathcal{B}||^2}\right)$ is the step-length. However, in many practical application, there are two major drawbacks in the implementations of Algorithm (1.4): the first is that, it requires in each iteration to computes two projections $P_{\mathcal{C}}$ and $P_{\mathcal{Q}}$, which depends heavily on the geometry of the sets $\mathcal{C}$ and $\mathcal{Q}$, these are extremely expensive operations and sometimes not even possible for a wide range of practical problems and the second is that, the step length depends on the information of the norm of $\mathcal{B}$, which is generally very hard to obtain in many practice.

By defining $\mathcal{C}$ and $\mathcal{Q}$ as the following sub level sets:

$$\mathcal{C} = \{x \in \mathcal{H}_1 : c(x) \leq 0\}, \quad \mathcal{Q} = \{t \in \mathcal{H}_2 : q(t) \leq 0\}, \tag{1.5}$$

where $c : \mathcal{H}_1 \to \mathbb{R}$ and $q : \mathcal{H}_2 \to \mathbb{R}$ are weakly lower semi-continuous and convex functions and the two half-spaces at points x_n by

$$\mathcal{C}_n = \{x \in \mathcal{H}_1 : c(x_n) \leq \langle \phi_n, x_n - x\rangle\}, \quad \mathcal{Q}_n = \{t \in \mathcal{H}_2 : q(\mathcal{B}x_n) \leq \langle \varphi_n, \mathcal{B}x_n - t\rangle\}, \tag{1.6}$$

with $\phi_n \in \partial c(x_n)$, $\varphi_n \in \partial q(\mathcal{B}x_n)$, $\mathcal{C} \subseteq \mathcal{C}_n$ and $\mathcal{Q} \subseteq \mathcal{Q}_n$ for each $n \geq 0$, Yang [59] proposed the relaxed version of the method (1.4), which suggests to replace the two arbitrary sets $\mathcal{C}$ and $\mathcal{Q}$ with the half-spaces $\mathcal{C}_n$ and $\mathcal{Q}_n$, respectively, so that the projections $P_{\mathcal{C}_n}$ and $P_{\mathcal{Q}_n}$ can easily be computed using their known closed-form expressions (see [5], Example 29.20).

On the other hand, some researchers have suggested some methods, which do not require the calculation of $||\mathcal{B}||$. One of such methods is that of Qu and Xiu [44], in which they adopted an Armijo-like step length and presented a modified version of the algorithm in [59]. In this light, the authors of the works in [18, 23, 49] subsequently proposed some algorithms with Armijo-like step lengths to solve problem (1.1). It has been noted that finding the step length that is appropriate in each iteration using Armijo-like step length involves multiple search procedures, which may leads to an inefficiency in the performance and computations of the algorithms. To mitigate this drawback, Dong et al. [21] proposed an adaptive relaxed algorithm for the problem (1.1), in which the authors adopted the simple ways of computing a monotonic step length in each iteration based on the information of the previous iterates. Similarly, very recently, Tan et al. [53] introduced another adaptive relaxed algorithm based on the non-monotonic step length technique.

However, various researchers attempt to construct some methods with fast convergence properties, since they are mostly required in various applications [12, 32]. In recent years, some authors developed various algorithms [50, 46, 4, 39, 45, 51, 42, 27, 58, 2] based on Polyak's inertial method [43], to improve their convergence rates. However, it has been noted in several instances that the speed of some methods with Polyak's one-step inertial term

$$x_n + \lambda(x_n - x_{n-1}), \quad \forall \lambda > 0,$$

appear to be slower than their corresponding non-inertial ones (see [6, 38] and the references therein). Thus some authors [37, 14, 19] suggested to employ the idea of the multi-step inertial technique, which could help to maintained the expected improvements in the speed of these schemes. Additionally, to improve the speed of the inertial algorithms, the idea of the general inertial technique with two inertial steps was introduced by Dong et al. [17], which includes the classical Polyak's inertial method as a special case. Some researchers incorporated the idea of the general inertial method to improve the performance of their methods with several real-world applications (see e.g., [35, 57]). Similarly, motivated by the idea of the multi-step inertial technique and that of the general inertial technique, Dong et al. [19] introduced the general multi-step inertial Krasnosel'skíí -Mann algorithm, which is formulated as follows:

$$\left\{\begin{array}{l} w_n = x_n + \sum_{k\in K_n} \gamma_{n,k}(x_{n-k} - x_{n-k-1}), \\ v_n = x_n + \sum_{k\in K_n} \delta_{n,k}(x_{n-k} - x_{n-k-1}), \\ x_{n+1} = (1-\alpha_n)w_n + \alpha_n T v_n, \quad \forall n \geq 1, \end{array}\right. \tag{1.7}$$

where $K_n \subseteq \{0, 1, 2, \cdots, n-1\}$, $\gamma_{n,k},\ \delta_{n,k} \in (-1, 2]$. They proved its weak convergence to a fixed point of a nonexpansive operator T based on the convergence of the Krasnosel'skíí-Mann algorithm with perturbations in a real Hilbert space. They numerically show that the scheme (1.7) is faster than some inertial methods in solving the problems considered in [19]. Additionally, for any two given points x_{n-1} and x_n for each $n \geq 1$, Mu and Peng [40] suggested the following alternated inertial term:

$$y_n = \begin{cases} x_n, & \text{if } n \text{ is even,} \\ x_n + \lambda_n(x_n - x_{n-1}), & \text{if } n \text{ is odd,} \end{cases} \tag{1.8}$$

which is a modification of the Polyak's inertial method. The advantage of the modified version in (1.8) is its ability to recover Fejér monotonicity property of its even subsequence in relation to the set of the solutions of a problem. This important property is usually lost in the case of the non-modified version. Very recently, some methods based on (1.8) for solving problem (1.1) were developed [21, 53, 48, 1]. Although the algorithms in [21, 53, 48] based on (1.8) were shown to achieve better computational efficiencies when their numerical results are compared with some existing methods on signal and image processing problems, but their weak convergence property was only obtained.

Additionally, in view of (1.3) and the fact that $\nabla g(x) = \mathcal{B}^*(I - P_{\mathcal{Q}})\mathcal{B}x$, it is not difficult to see that all the aforementioned methods for solving problem (1.1), such as those in [50, 7, 59, 44, 18, 23, 49, 21, 53, 46, 39, 45, 48, 1] are hybrid steepest-types with the directions $d_n = -\nabla g_n(x_n)$ at a point x_n. However, as noted from [33], the accelerated versions of these methods may be constructed when considered with the following conjugate gradient-like direction (1.9) or the three-term conjugate gradient-like direction (1.10) (see [31, 30]):

$$d_n = -\nabla g_n(x_n) + \varsigma_n^{(1)} d_{n-1} \tag{1.9}$$

and

$$d_n = -\nabla g_n(x_n) + \varsigma_n^{(1)} d_{n-1} - \varsigma_n^{(2)} s_n, \forall n \geq 1, \tag{1.10}$$

respectively, where, for each $i = 1, 2$, $\varsigma_n^{(i)} \in [0, \infty)$ and $s_n \in \mathcal{H}_1$ is an arbitrary point. As numerically shown in [33, 31, 30], provided that, for each $i = 1, 2$, $\lim_{n\to\infty} \varsigma_n^{(i)} = 0$ and $\{s_n\}$ is bounded, the hybrid gradient method with the direction (1.10) is faster than its variant with the direction (1.9). In the light of this, some authors improved their iterative methods by combining them with either of the directions (1.9) or (1.10) for different problems (see [26, 16, 3, 36] and the references therein). Recently, motivated by the self-adaptive relaxed algorithm [60], Polyak's one-step inertial method [43] and the conjugate gradient-like direction (1.9), Che et al. [11] proposed the accelerated relaxed algorithm for the problem (1.1) in finite-dimensional real Hilbert spaces. Although the proposed algorithm in [11] with the conjugate gradient-like direction (1.9) has achieved some good performance on signal and image restoration problems, but it is noted that its convergence results heavely rely on the conditions that, for any sequence $\{x_n\}$ generated by the their algorithm, the sequences $\{(I - P_{\mathcal{C}_n})x_n\}$ and $\{(I - P_{\mathcal{Q}_n})\mathcal{B}x_n\}$ are bounded. These are very restrictive assumptions and it would be of great interest to dispense them.

Motivated and inspired by the results in [21, 53, 37, 17, 40, 33], we first develop an alternated inertial Halpern-type relaxed $\mathcal{CQ}$ algorithm with perturbations (AiHRAP), which employs the monotonic self-adaptive step length criterion that does not require any information about the norm of the operator or the use of a line search procedure. Moreover, we establish its strong convergence to a minimum-norm solution of problem (1.1) in infinite-dimensional real Hilbert spaces. Further, we introduce two extensions of AiHRAP: the first is an alternated and multi-step inertial Halpern-type relaxed $\mathcal{CQ}$ algorithm (AMiHRA), which to the best of our knowledge is the most general inertial method in the literature that involves three steps of the recent improvements of the classical inertial method, one of which is the alternated inertial step [40], while the others are the multi-step inertial steps [37], and the second is an accelerated alternated and multi-step inertial Halpern-type relaxed algorithm (AAMiHRA) that combines the three term conjugate gradient-like direction [33] and two steps of the aforementioned improved versions of the inertial term with the monotonic self-adaptive step length criterion. Moreover, we analyse their applications on classification problems for some real-world datasets based on the extreme learning machine (ELM) with the ℓ_1-regularization approach (that is, the Lasso model) and the $\ell_1 - \ell_2$ hybrid regularization approach. Furthermore, we investigate their performance in solving constrained minimization problems in infinite-dimensional Hilbert spaces.

2. Preliminaries

In this work, we use $x_n \to x^*$ (resp., $x_n \rightharpoonup x^*$) to represent the strong (resp., weak) convergence of a sequence $\{x_n\}$ to a point x^*. For any $x, y \in \mathcal{H}$ and $\lambda \in [0, 1]$, we require the following identities:

$$||x + y||^2 = ||x||^2 + ||y||^2 + 2\langle x, y\rangle \tag{2.11}$$

and

$$||\lambda x + (1 - \lambda)y||^2 = \lambda||x||^2 + (1 - \lambda)||y||^2 - \lambda(1 - \lambda)||x - y||^2. \tag{2.12}$$

Definition 2.1 ([5]). *Let $\mathcal{T} : \mathcal{H} \to \mathcal{H}$ be a mapping. Then $\mathcal{T}$ is called*

(1) K-Lipschitz continuous with $K > 0$ if

$$||\mathcal{T}x - \mathcal{T}y|| \leq K||x - y||, \quad \forall x, y \in \mathcal{H}; \tag{2.13}$$

(2) nonexpansive if (2.13) holds with $K = 1$;

(3) firmly nonexpansive if

$$||\mathcal{T}x - \mathcal{T}y|| \leq \langle x - y, \mathcal{T}x - \mathcal{T}y\rangle, \quad \forall x, y \in \mathcal{H}. \tag{2.14}$$

For any $x \in \mathcal{H}$ and $y \in \mathcal{C}$, we have the following properties (see [25]):

$$\langle x - P_{\mathcal{C}}x, P_{\mathcal{C}}x - y\rangle \geq 0, \tag{2.15}$$

equivalently,

$$||x - P_{\mathcal{C}}x||^2 + ||y - P_{\mathcal{C}}x||^2 \leq ||x - y||^2. \tag{2.16}$$

Remark 2.1. *It is commonly known that $I - P_{\mathcal{C}}$ satisfies the inequality (2.14) (see [56]).*

Definition 2.2 ([5]). *Let $f : \mathcal{H} \to (-\infty, +\infty]$ be a convex and proper function. Then:*

(1) f is said to be (weakly) lower semi-continuous (w-lsc) if for any sequence $x_n \in \mathcal{H}$ such that ($x_n \rightharpoonup x^*$) $x_n \to x^*$ as $n \to \infty$, we have

$$\liminf_{n\to\infty} f(x_n) \geq f(x^*). \tag{2.17}$$

(2) $\partial f(x)$ is known as the subdifferential of f at a point x, which is defined by

$$\partial f(x) := \{v \in \mathcal{H} : \langle v, y - x\rangle + f(x) \leq f(y), \forall y \in \mathcal{H}\}.$$

An element $v \in \partial f(x)$ is called a subgradient of f at x.

Lemma 2.1 ([56, 9]). *Let $\tau > 0$ and $x^* \in \mathcal{H}_1$. The point x^* solves problem (1.1) if and only if it solves the fixed point problem:*

$$x^* = P_{\mathcal{C}}(I - \tau\mathcal{B}^*(I - P_{\mathcal{Q}})\mathcal{B})x^*.$$

Lemma 2.2 ([28]). *Let $\{x_n\}$ be a sequence of nonnegative real numbers such that $\forall n \geq 1$,*

$$x_{n+1} \leq (1 - \beta_n)x_n + \beta_n\Gamma_n,$$

$$x_{n+1} \leq x_n - \chi_n + \Phi_n, \forall n \geq 1,$$

where $\beta_n \in (0, 1)$, $\chi_n \in [0, +\infty)$ and Γ_n, $\Phi_n \in (-\infty, +\infty)$ such that

(B1) $\lim\limits_{n\to\infty} \beta_n = 0$ and $\sum_{n=1}^{\infty} \beta_n = \infty$;
(B2) $\lim\limits_{n\to\infty} \Phi_n = 0$;
(B3) $\lim\limits_{j\to\infty} \chi_{n_j} = 0$ implies that $\limsup\limits_{r\to\infty} \Gamma_{n_j} \leq 0$ for any subsequence $\{n_j\}$ of $\{n\}$,

Then $\lim\limits_{n\to\infty} x_n = 0$.

3. Main results

3.1. **Alternated Inertial Halpern-type Relaxed Algorithm with Perturbations.** In this part, we introduce the alternated inertial Halpern-type relaxed algorithm with perturbations and analyse its strong convergence to the minimum-norm solution of the problem (1.1) in real Hilbert spaces. For its construction, we define g_n, $\mathcal{C}$, $\mathcal{Q}$, $\mathcal{C}_n$ and $\mathcal{Q}_n$ as in the equations (1.3), (1.5) and (1.6), respectively. Moreover, to establish its convergence, we require the conditions in the following assumption:

Assumption 1:

($A1$) The solutions' set of problem (1.1) is denoted by $\Omega \neq \emptyset$.
($A2$) $c : \mathcal{H}_1 \to \mathbb{R}$ and $q : \mathcal{H}_2 \to \mathbb{R}$ are respectively convex, subdifferentiable and weakly lower semicontinuous functions on $\mathcal{H}_1$ and $\mathcal{H}_2$.
($A3$) For any $x \in \mathcal{H}_1$ and $y \in \mathcal{H}_2$, at least one subgradient $\phi \in \partial c(x)$ and $\varphi \in \partial q(y)$ are obtainable and the subdifferential operators ∂c and ∂q are bounded on bounded sets.
($A4$) Let $\tau_1 > 0$, $\varepsilon > 0$, $\rho \in (0, \frac{1}{\varepsilon})$, $\delta_n \in (0, 1)$ such that $\lim\limits_{n\to\infty} \delta_n = 0$ and $\sum_{n=0}^{\infty} \delta_n = +\infty$.

Algorithm 1 Alternated inertial Halpern-type Relaxed $\mathcal{CQ}$ Algorithm with Perturbations (AiHRAP)

Initialization: Take $\tau_1,\ \varepsilon,\ \rho$ and $\{\delta_n\}$ such that the conditions $(A4)$ of Assumption 1 holds. Select $\lambda_n \in [0,+\infty)$, $u \in \mathcal{C}$, $x_0,\ x_1 \in \mathcal{H}_1$ and set $n = 1$.

Step 1. Compute

$$y_n = \begin{cases} x_n, & \text{if } n \text{ is even}, \\ x_n + \lambda_n(x_n - x_{n-1}), & \text{if } n \text{ is odd}. \end{cases} \tag{3.18}$$

Step 2. Compute

$$h_n = P_{\mathcal{C}_n}(y_n - \rho\tau_n \nabla g_n(y_n) + e_1(y_n)).$$

If $h_n = y_n$, then, stop the iteration and $h_n \in \Omega$, else, go to Step 3.

Step 3. Compute

$$m_n = P_{\mathcal{C}_n}(y_n - \rho\tau_n \nabla g_n(h_n) + e_2(y_n)).$$

Step 4. Compute

$$x_{n+1} = \delta_n u + (1-\delta_n)m_n$$

and update the step-length τ_{n+1} by

$$\tau_{n+1} = \begin{cases} \min\left\{\frac{\varepsilon||y_n - h_n||}{||\nabla g_n(y_n) - \nabla g_n(h_n)||},\ \tau_n\right\}, & \text{if } \nabla g_n(y_n) \neq \nabla g_n(h_n), \\ \tau_n, & \text{otherwise}. \end{cases} \tag{3.19}$$

Set $n := n+1$ and go back to Step 1.

Remark 3.2. *In Algorithm 1, for all $n \geq 1$, we select the inertial parameter λ_n as follows;*

$$\lambda_n = \begin{cases} \min\left\{\frac{\xi_n}{||x_n - x_{n-1}||^2},\ \eta_1\right\}, & \text{if } x_n \neq x_{n-1}, \\ \eta_1, & \text{otherwise}, \end{cases} \tag{3.20}$$

where $\xi_n \in [0,+\infty)$ such that $\lim_{n\to\infty} \frac{\xi_n}{\delta_n} = 0$ and $\eta_1 > 0$. Moreover, for the analysis of the convergence of Algorithm 1, we provide the following additional assumption:

Assumption 2: Assume that, for each $i = 1, 2$, the sequence of perturbations $\{e_i(y_n)\}$ satisfies

$$\lim_{n\to\infty} \frac{||e_i(y_n)||}{\delta_n} = 0. \tag{3.21}$$

Remark 3.3. *It appears from Algorithm 1 that*

$$m_n = P_{\mathcal{C}_n}(y_n - \rho\tau_n \nabla g_n(h_n)) + \bar{e}_2(y_n), \forall n \geq 1 \tag{3.22}$$

so that

$$\begin{aligned} ||\bar{e}_2(y_n)|| &= ||P_{\mathcal{C}_n}(y_n - \rho\tau_n \nabla g_n(h_n) + e_2(y_n)) - P_{\mathcal{C}_n}(y_n - \rho\tau_n \nabla g_n(h_n))|| \\ &\leq ||e_2(y_n)||. \end{aligned} \tag{3.23}$$

Combining (3.21) and (3.23), we have

$$\lim_{n\to\infty} \frac{||\bar{e}_2(y_n)||}{\delta_n} = 0. \tag{3.24}$$

In the first, we validate the stopping criterion of Algorithm 1 in the following remark.

Remark 3.4. *If we let $h_n = y_n$ in Algorithm 1, then we see that*

$$h_n = P_{\mathcal{C}_n}(h_n - \rho\tau_n \nabla g_n(h_n) + e_1(h_n)), \forall n \geq 1,$$

which implies that $h_n \in \mathcal{C}_n$. Thus, by the means of Lemma 2.1, we have $Bh_n \in \mathcal{Q}_n$. Together with (1.5) and (1.6), we obtain that $h_n \in \mathcal{C}$ and $Bh_n \in \mathcal{Q}$. Therefore, $h_n \in \Omega$.

Lemma 3.3. *Suppose that $\{\tau_n\}$ is a sequence of step lengths generated by (3.19). Then it is well defined and $\tau_n \geq \frac{\varepsilon}{||B||^2}$ for all $n \geq 1$.*

Proof. By the lipschitz contuinity of ∇g_n with constant $||B||^2$, we obtain

$$\frac{\varepsilon||y_n - h_n||}{||\nabla g_n(y_n) - \nabla g_n(h_n)||} \geq \frac{\varepsilon||y_n - h_n||}{||B||^2||y_n - h_n||} = \frac{\varepsilon}{||B||^2}.$$

In view of this and (3.19), one sees that $\tau_{n+1} \geq \min\{\tau_n, \frac{\varepsilon}{||B||^2}\}$. By induction, we obtain that $\tau_n \geq \min\{\tau_1, \frac{\varepsilon}{||B||^2}\}$. It is also seen from (3.19) that $\tau_{n+1} \leq \tau_n$ for all $n \in \mathbb{N}$. In view of the monotonicity and the existence of the lower bound of the sequence $\{\tau_n\}$, we obtain that $\lim\limits_{n\to\infty} \tau_n$ exists. Since $\min\{\tau_1, \frac{\varepsilon}{||B||^2}\}$ is a lower bound of the sequence $\{\tau_n\}$, we can find $\tau > 0$ such that $\lim\limits_{n\to\infty} \tau_n = \tau$. This completes the proof. ☐

Next, we establish that an even subsequence $\{x_{2n}\}$ of $\{x_n\}$ by Algorithm 1 is bounded.

Lemma 3.4. *Let $\{x_n\}$ be a sequence produced by Algorithm 1. Then, for any point $z \in \Omega$, an even subsequence $\{||x_{2n} - z||\}$ of $\{||x_n - z||\}$ is bounded.*

Proof. Let $z \in \Omega$. Then $Bz \in \mathcal{Q}_n$ and, consequently, $\nabla g_n(z) = B^*(I - P_{\mathcal{Q}_n})Bz = 0$. Therefore, together with the fact that $I - P_{\mathcal{Q}_n}$ satisfies (2.14), we have

$$\begin{aligned}
\langle \nabla g_n(h_n), h_n - z\rangle &= \langle B^*(I - P_{\mathcal{Q}_n})Bh_n - B^*(I - P_{\mathcal{Q}_n})Bz, h_n - z\rangle \\
&= \langle (I - P_{\mathcal{Q}_n})Bh_n - (I - P_{\mathcal{Q}_n})Bz, Bh_n - Bz\rangle \\
&\geq ||(I - P_{\mathcal{Q}_n})Bh_n||^2 \\
&= 2g_n(h_n), \forall n \geq 1. \qquad (3.25)
\end{aligned}$$

Letting $p_n = P_{\mathcal{C}_n}(y_n - \rho\tau_n\nabla g_n(h_n))$, it follows from the inequalities (2.16) and (3.25) that

$$\begin{aligned}
||p_n - z||^2 &\leq ||P_{\mathcal{C}_n}(y_n - \rho\tau_n\nabla g_n(h_n)) - z||^2 \\
&\leq ||y_n - \rho\tau_n\nabla g_n(h_n) - z||^2 - ||y_n - \rho\tau_n\nabla g_n(h_n) - p_n||^2 \\
&= ||y_n - z||^2 - ||y_n - p_n||^2 - 2\rho\tau_n \langle \nabla g_n(h_n), y_n - z\rangle \\
&+ 2\rho\tau_n \langle \nabla g_n(h_n), y_n - p_n\rangle \\
&\leq ||y_n - z||^2 - ||y_n - p_n||^2 - 4\rho\tau_n g_n(h_n) \\
&- 2\rho\tau_n \langle \nabla g_n(h_n), p_n - h_n\rangle . \qquad (3.26)
\end{aligned}$$

Now, we estimate the rightmost term of (3.26) as follows:
We noticed from (2.11) that

$$||y_n - h_n||^2 + ||h_n - p_n||^2 - ||y_n - p_n||^2 = 2\langle y_n - h_n, p_n - h_n\rangle . \qquad (3.27)$$

By the fact that $p_n \in \mathcal{C}_n$, we obtain from (3.19), the property (2.15) and the mean value inequality that

$$\begin{aligned}
2\langle y_n - h_n, p_n - h_n\rangle &= 2\langle y_n - \rho\tau_n \nabla g_n(y_n) + e_1(y_n) - h_n, p_n - h_n\rangle \\
&+ 2\rho\tau_n \langle \nabla g_n(y_n) - \nabla g_n(h_n), p_n - h_n\rangle \\
&+ 2\rho\tau_n \langle \nabla g_n(h_n), p_n - h_n\rangle \\
&- 2\langle e_1(y_n), p_n - h_n\rangle \\
&\le 2\rho\tau_n ||\nabla g_n(y_n) - \nabla g_n(h_n)|| ||p_n - h_n|| \\
&+ 2\rho\tau_n \langle \nabla g_n(h_n), p_n - h_n\rangle \\
&+ 2||e_1(y_n)|| ||p_n - h_n|| \\
&\le \Big(\frac{\varepsilon\rho\tau_n}{\tau_{n+1}} + ||e_1(y_n)||\Big)\big(||y_n - h_n||^2 + ||p_n - h_n||^2\big) \\
&+ ||e_1(y_n)|| + 2\rho\tau_n \langle \nabla g_n(h_n), p_n - h_n\rangle . \qquad (3.28)
\end{aligned}$$

Combining (3.27) and (3.28), we deduce that

$$\begin{aligned}
2\rho\tau_n \langle \nabla g_n(h_n), p_n - h_n\rangle &\ge \Big(1 - \Big(\frac{\varepsilon\rho\tau_n}{\tau_{n+1}} + ||e_1(y_n)||\Big)\Big)\big(||y_n - h_n||^2 + ||p_n - h_n||^2\big) \\
&- ||e_1(y_n)|| - ||y_n - p_n||^2. \qquad (3.29)
\end{aligned}$$

In view of the inequalities (3.26), (3.29) and Lemma 3.3, one sees that

$$\begin{aligned}
||p_n - z||^2 &\le ||y_n - z||^2 - \frac{4\rho\varepsilon}{||B||^2} g_n(h_n) + ||e_1(y_n)|| \\
&- \rho_n\big(||y_n - h_n||^2 + ||p_n - h_n||^2\big), \qquad (3.30)
\end{aligned}$$

where

$$\rho_n = \Big(1 - \Big(\frac{\varepsilon\rho\tau_n}{\tau_{n+1}} + ||e_1(y_n)||\Big)\Big). \qquad (3.31)$$

Note that, for any $\varepsilon > 0$ and $\rho \in (0, \frac{1}{\varepsilon})$, we immediately see, from Lemma 3.3, Assumption 2 and equation (3.31), that there exists $\rho^* > 0$ such that $\lim_{n\to\infty} \rho_n = \rho^*$, where

$$\rho^* = (1 - \varepsilon\rho). \qquad (3.32)$$

Thus we can find a positive number R such that $\rho_n > 0$ for all $n \ge R$. Together with (3.30) and the definition of m_n in Algorithm 1, we see that

$$\begin{aligned}
||m_n - z||^2 &= ||p_n + \bar{e}_2(y_n) - z||^2 \\
&\le (1 + ||\bar{e}_2(y_n)||)||p_n - z||^2 + ||\bar{e}_2(y_n)|| + ||\bar{e}_2(y_n)||^2 \\
&\le \big(1 + ||\bar{e}_2(y_n)||\big)||y_n - z||^2 + \vartheta_n - \Theta_n \\
&\le \big(1 + ||\bar{e}_2(y_n)||\big)||y_n - z||^2 + \vartheta_n, \forall n \ge R, \qquad (3.33)
\end{aligned}$$

where

$$\Theta_n = \big(1 + ||\bar{e}_2(y_n)||\big)\Big(\frac{4\rho\varepsilon}{||B||^2} g_n(h_n) + \rho_n\big(||y_n - h_n||^2 + ||p_n - h_n||^2\big)\Big)$$

and

$$\vartheta_n = \big(1 + ||\bar{e}_2(y_n)||\big)||e_1(y_n)|| + ||\bar{e}_2(y_n)|| + ||\bar{e}_2(y_n)||^2.$$

Using the convexity of $||\cdot||^2$, it follows from (3.33) that

$$\begin{aligned} ||x_{n+1}-z||^2 &\le \delta_n||u-z||^2+(1-\delta_n)\big(1+||\bar{e}_2(y_n)||\big)||y_n-z||^2 \\ &\quad +(1-\delta_n)(\vartheta_n-\Theta_n). \end{aligned} \tag{3.34}$$

In view of (3.18) and taking $n+1=2n+1$ in (3.34), we see that

$$\begin{aligned} ||x_{2n+1}-z||^2 &\le \delta_{2n}||u-z||^2+(1-\delta_{2n})\big(1+||\bar{e}_2(y_{2n})||\big)||x_{2n}-z||^2 \\ &\quad +(1-\delta_{2n})(\vartheta_{2n}-\Theta_{2n}) \end{aligned} \tag{3.35}$$

and

$$\begin{aligned} ||y_{2n+1}-z||^2 &\le (1+\lambda_{2n+1})||x_{2n+1}-z||^2-\lambda_{2n+1}||x_{2n}-z||^2 \\ &\quad +\lambda_{2n+1}(1+\lambda_{2n+1})||x_{2n+1}-x_{2n}||^2. \end{aligned} \tag{3.36}$$

Combining (3.35) and (3.36) for $n+1=2n+2$ in (3.34), we deduce that

$$\begin{aligned} ||x_{2n+2}-z||^2 &\le \delta_{2n+1}||u-z||^2+(1-\delta_{2n+1})\big(1+||\bar{e}_2(y_{2n+1})||\big)||y_{2n+1}-z||^2 \\ &\quad +(1-\delta_{2n+1})(\vartheta_{2n+1}-\Theta_{2n+1}) \\ &\le \delta_{2n+1}||u-z||^2+\big(1+||\bar{e}_2(y_{2n+1})||\big)(1+\lambda_{2n+1})\Big(\delta_{2n}||u-z||^2 \\ &\quad +(1-\delta_{2n})\big(1+||\bar{e}_2(y_{2n})||\big)||x_{2n}-z||^2+(1-\delta_{2n})(\vartheta_{2n}-\Theta_{2n})\Big) \\ &\quad -\lambda_{2n+1}\big(1+||\bar{e}_2(y_{2n+1})||\big)||x_{2n}-z||^2+(1-\delta_{2n+1})(\vartheta_{2n+1}-\Theta_{2n+1}) \\ &\quad +\lambda_{2n+1}\big(1+||\bar{e}_2(y_{2n+1})||\big)(1+\lambda_{2n+1})||x_{2n+1}-x_{2n}||^2 \\ &\le (1-\delta_{2n})\big(1+||\bar{e}_2(y_{2n})||\big)\big(1+||\bar{e}_2(y_{2n+1})||\big)(1+\lambda_{2n+1})||x_{2n}-z||^2 \\ &\quad +\big(1+||\bar{e}_2(y_{2n+1})||\big)(1+\lambda_{2n+1})\big(2\delta_{2n}||u-z||^2+(1-\delta_{2n})(\vartheta_{2n}-\Theta_{2n}) \\ &\quad +(1-\delta_{2n+1})(\vartheta_{2n+1}-\Theta_{2n+1})+\lambda_{2n+1}||x_{2n+1}-x_{2n}||^2\big). \end{aligned} \tag{3.37}$$

Using (3.31), (3.32) and the fact that for any $\varepsilon>0$, $\rho\in(0,\frac{1}{\varepsilon})$, we find from (3.37) that

$$\begin{aligned} ||x_{2n+2}-z||^2 &\le (1-\delta_{2n})||x_{2n}-z||^2+\frac{1}{\big(1+||\bar{e}_2(y_{2n})||\big)}\Big(2\delta_{2n}||u-z||^2+(1-\delta_{2n})\vartheta_{2n} \\ &\quad +(1-\delta_{2n+1})\vartheta_{2n+1}+\lambda_{2n+1}||x_{2n+1}-x_{2n}||^2\Big),\ \forall\ n\ge R. \end{aligned} \tag{3.38}$$

Taking

$$\begin{aligned} M=\sup_{n\ge 1}&\frac{1}{\big(1+||\bar{e}_2(y_{2n})||\big)}\Big(2||u-z||^2+\frac{(1-\delta_{2n})}{\delta_{2n}}\vartheta_{2n}+\frac{\lambda_{2n+1}}{\delta_{2n}}||x_{2n+1}-x_{2n}||^2 \\ &+\frac{(1-\delta_{2n+1})}{\delta_{2n}}\vartheta_{2n+1}\Big), \end{aligned}$$

then, by (3.38) and the condition $(A4)$, we obtain that

$$\begin{aligned} ||x_{2n+2}-z||^2 &\le (1-\delta_{2n})||x_{2n}-z||^2+\delta_{2n}M \\ &\le \max\big\{||x_{2n}-z||^2, M\big\} \\ &\ \ \vdots \\ &\le \max\big\{||x_0-z||^2, M\big\}, \forall n\ge R. \end{aligned} \tag{3.39}$$

By Remark 3.2, Assumption 2, the condition $(A4)$ and the inequality (3.39), we obtain that, for any $z\in\Omega$, the even subsequence $\{||x_{2n}-z||\}$ of $\{||x_n-z||\}$ produced by Algorithm 1 is bounded. Consequently, the even subsequence $\{x_{2n}\}$ of $\{x_n\}$ generated by Algorithm 1 is bounded. This completes the proof. $\square$

Next is to state and prove the following strong convergence theorem for Algorithm 1:

Theorem 3.1. *Let the conditions of Assumptions* 1, 2 *and Remark* 3.2 *hold, and* $\{x_n\}$ *be a sequence generated by Algorithm* 1*. Then,* $\{x_n\}$ *converges strongly to a point* $z^* \in \Omega$*, where* $z^* = P_\Omega 0$*.*

Proof. Let $z \in \Omega$. Then, by (2.11) and (3.33), we get

$$\begin{aligned} ||x_{n+1}-z||^2 &= \delta_n^2||u-z||^2 + (1-\delta_n)^2||m_n-z||^2 + 2\delta_n(1-\delta_n)\langle m_n - z, u-z\rangle \\ &\le (1-\delta_n)\big(1+||\bar{e}_2(y_n)||\big)||y_n-z||^2 + (1-\delta_n)\vartheta_n + \delta_n^2||u-z||^2 \\ &\quad + 2\delta_n(1-\delta_n)\langle m_n-z, u-z\rangle . \end{aligned} \tag{3.40}$$

Similar arguments used in deriving (3.35) lead to obtain from (3.40) that

$$\begin{aligned} ||x_{2n+1}-z||^2 &\le (1-\delta_{2n})\big(1+||\bar{e}_2(y_{2n})||\big)||x_{2n}-z||^2 + (1-\delta_{2n})\vartheta_{2n} \\ &\quad + \delta_{2n}^2||u-z||^2 + 2\delta_{2n}(1-\delta_{2n})\langle m_{2n}-z, u-z\rangle . \end{aligned} \tag{3.41}$$

Connecting (3.36) and (3.41) for $n+1 = 2n+2$ in (3.40) and following same lines of the proof of (3.38), one finds that

$$\begin{aligned} ||x_{2n+2}-z||^2 &\le \big(1+||\bar{e}_2(y_{2n+1})||\big)||y_{2n+1}-z||^2 + (1-\delta_{2n+1})\vartheta_{2n+1} \\ &\quad + \delta_{2n+1}^2||u-z||^2 + 2\delta_{2n+1}(1-\delta_{2n+1})\langle m_{2n+1}-z, u-z\rangle \\ &\le (1-\delta_{2n})||x_n-z||^2 + \frac{1}{\big(1+||\bar{e}_2(y_n)||\big)}\Big(2\delta_{2n}^2||u-z||^2 + (1-\delta_{2n})\vartheta_{2n} \\ &\quad + \lambda_{2n+1}||x_{2n+1}-x_{2n}||^2 + 2\delta_{2n}(1-\delta_{2n})\langle m_{2n}-z, u-z\rangle\Big) \\ &\quad + \frac{2\delta_{2n+1}(1-\delta_{2n+1})}{\big(1+||\bar{e}_2(y_{2n})||\big)\big(1+||\bar{e}_2(y_{2n+1})||\big)(1+\lambda_{2n+1})}\langle m_{2n+1}-z, u-z\rangle \\ &\quad + (1-\delta_{2n+1})\vartheta_{2n+1}. \end{aligned} \tag{3.42}$$

Without loss of generality, using the condition $(A4)$ and Assumption 2, we assume that $r, s > 0$ exist such that, for all $n \ge 1$,

$$\frac{4(1-\delta_n)\rho\varepsilon}{\big(1+||\bar{e}_2(y_{n-1})||\big)(1+\lambda_n)||\mathcal{B}||^2} \ge r, \quad \frac{(1-\delta_n)\rho_n}{\big(1+||\bar{e}_2(y_{n-1})||\big)(1+\lambda_n)} \ge s.$$

In view of (3.37) and (3.42), one observes that

$$||x_{2n+2}-z||^2 \le ||x_{2n}-z||^2 - \chi_{2n} + \Phi_{2n}$$

and

$$||x_{2n+2}-z||^2 \le (1-\delta_{2n})||x_{2n}-z||^2 + \delta_{2n}\Gamma_{2n}, \tag{3.43}$$

where

$$\begin{aligned} \chi_{2n} &= p\big(g_{2n}(h_{2n}) + g_{2n+1}(h_{2n+1})\big) + q\big(||y_{2n}-h_{2n}||^2 + ||p_{2n}-h_{2n}||^2 \\ &\quad + ||y_{2n+1}-h_{2n+1}||^2 + ||p_{2n+1}-h_{2n+1}||^2\big), \end{aligned}$$

$$\begin{aligned} \Phi_{2n} =& \frac{1}{\big(1+||\bar{e}_2(y_{2n})||\big)}\big(2\delta_{2n}||u-z||^2 + (1-\delta_{2n})\vartheta_{2n} + (1-\delta_{2n+1})\vartheta_{2n+1} \\ & + \lambda_{2n+1}||x_{2n+1}-x_{2n}||^2\big) \end{aligned}$$

and

$$\Gamma_{2n} = \frac{1}{(1+||\bar{e}_2(y_n)||)}\Big(2\delta_{2n}||u-z||^2 + \frac{(1-\delta_{2n})}{\delta_{2n}}\vartheta_{2n} + \frac{\lambda_{2n+1}}{\delta_{2n}}||x_{2n+1}-x_{2n}||^2$$
$$+ 2(1-\delta_{2n})\langle m_{2n}-z, u-z\rangle\Big)$$
$$+ \frac{2\delta_{2n+1}(1-\delta_{2n+1})}{\delta_{2n}(1+||\bar{e}_2(y_{2n})||)(1+||\bar{e}_2(y_{2n+1})||)(1+\lambda_{2n+1})}\langle m_{2n+1}-z, u-z\rangle$$
$$+ \frac{(1-\delta_{2n+1})}{\delta_{2n}}\vartheta_{2n+1}.$$

Using condition $(A4)$, Assumptions 2 and Remark 3.2, we find that $\lim_{n\to\infty}\Phi_{2n} = 0$. Thus, to apply Lemma 2.2, it remains only to show that, for any subsequence $\{\chi_{2n_j}\}$ of $\{\chi_{2n}\}$, the following is true:

$$\lim_{j\to\infty}\chi_{2n_j} = 0 \Longrightarrow \limsup_{j\to\infty}\Gamma_{2n_j} \leq 0.$$

Now, suppose that $\{\chi_{2n_j}\}$ is a subsequence of $\{\chi_{2n}\}$ such that $\lim_{j\to\infty}\chi_{2n_j} = 0$. Then, in view of (3.31), the condition $(A4)$, Assumption 2 and the fact that $\lim_{n\to\infty}\rho_n = \rho^* > 0$, we obtain that

$$\lim_{j\to\infty}||x_{2n_j}-h_{2n_j}|| = 0, \quad \lim_{j\to\infty}||p_{2n_j}-h_{2n_j}|| = 0,$$
$$\lim_{j\to\infty}||y_{2n_j+1}-h_{2n_j+1}|| = 0, \quad \lim_{j\to\infty}||p_{2n_j+1}-h_{2n_j+1}|| = 0,$$
$$\lim_{j\to\infty}g_{2n_j}(h_{2n_j}) = 0 \Longleftrightarrow \lim_{j\to\infty}||(I-P_{\mathcal{Q}_{2n_j}})\mathcal{B}h_{2n_j}||^2 = 0$$

and

$$\lim_{j\to\infty}g_{2n_j+1}(h_{2n_j+1}) = 0 \Longleftrightarrow \lim_{j\to\infty}||(I-P_{\mathcal{Q}_{2n_j+1}})\mathcal{B}h_{2n_j+1}||^2 = 0. \tag{3.44}$$

Since an even subsequence $\{x_{2n}\}$ of $\{x_n\}$ is bounded, it follows that there exists a subsequence $\{x_{2n_j}\}$ of $\{x_{2n}\}$ converging weakly to a point say x^*. Then the condition $(A3)$ of Assumption 1 guarantees the existence of a constant $\varrho > 0$ such that $||\varphi_{2n_j}|| \leq \varrho$. Together with the definition of $\mathcal{Q}_{2n_j}$, $P_{\mathcal{Q}_{2n_j}}\mathcal{B}h_{2n_j} \in \mathcal{Q}_{2n_j}$ and the results in (3.44), we have

$$\begin{aligned} q(\mathcal{B}h_{2n_j}) &\leq \Big\langle \varphi_{2n_j}, \mathcal{B}h_{2n_j} - P_{\mathcal{Q}_{2n_j}}\mathcal{B}h_{2n_j}\Big\rangle \\ &\leq ||\varphi_{2n_j}||||\mathcal{B}h_{2n_j} - P_{\mathcal{Q}_{2n_j}}\mathcal{B}h_{2n_j}|| \\ &\leq \varrho||(I-P_{\mathcal{Q}_{2n_j}})\mathcal{B}h_{2n_j}||^2 \to 0 \ \ as \ \ j\to\infty. \end{aligned} \tag{3.45}$$

So, It is not difficult to see from the weakly lower semicontuinity of q and (3.45) that

$$q(\mathcal{B}x^*) \leq \liminf_{j\to\infty} q(\mathcal{B}h_{2n_j}) \leq 0, \tag{3.46}$$

which implies that $\mathcal{B}x^* \in \mathcal{Q}$.

Similarly, the boundedness of ∂c on bounded sets also implies the existence of $\sigma > 0$, such that $||\phi_{2n_j}|| \leq \sigma$. From the definition of $\mathcal{C}_{2n_j}$, $p_{2n_j} \in \mathcal{C}_{2n_j}$ and (3.44), we see that

$$\begin{aligned} c(h_{2n_j}) &\leq \langle \phi_{2n_j}, h_{2n_j} - p_{2n_j}\rangle \\ &\leq ||\phi_{2n_j}||||h_{2n_j} - p_{2n_j}|| \\ &\leq \sigma||h_{2n_j} - p_{2n_j}|| \to 0 \ \ as \ \ j\to\infty. \end{aligned} \tag{3.47}$$

Using similar arguments used in deriving (3.46), one obtains that $c(x^*) \leq 0$, showing that $x^* \in \mathcal{C}$. Then the conclusion that $x^* \in \Omega$ is reached, which implies generally that $\omega_w(x_n) \subset \Omega$

since the choice of x^* was arbitrarily. From the definition of m_{2n_j} in Algorithm 1, (3.24) and (3.44), we, respectively, see that

$$\lim_{j\to\infty}||m_{2n_j} - p_{2n_j}|| = 0 \tag{3.48}$$

and

$$||x_{2n_j} - p_{2n_j}|| \le ||x_{2n_j} - h_{2n_j}|| + ||h_{2n_j} - p_{2n_j}|| \to 0 \text{ as } j \to \infty. \tag{3.49}$$

Combining (3.48) and (3.49), one finds that

$$\lim_{j\to\infty}||m_{2n_j} - x_{2n_j}|| = 0. \tag{3.50}$$

In view of the definition of x_{2n_j+1} in Algorithm 1, (3.50) and the condition $(A4)$, we deduce that

$$||x_{2n_j+1} - x_{2n_j}|| \le \delta_{2n_j}||u - x_{2n_j}|| + (1-\delta_{2n_j})||m_{2n_j} - x_{2n_j}|| \to 0 \text{ as } j \to \infty. \tag{3.51}$$

Similarly, by Remark 3.2, we get

$$||y_{2n_j+1} - x_{2n_j+1}|| \le \lambda_{2n_j+1}||x_{2n_j+1} - x_{2n_j}|| \to 0 \text{ as } j \to \infty. \tag{3.52}$$

It is also, respectively, seen from (3.44), (3.22) and (3.24) that

$$||y_{2n_j+1} - p_{2n_j+1}|| \le ||y_{2n_j+1} - h_{2n_j+1}|| + ||h_{2n_j+1} - p_{2n_j+1}|| \to 0 \text{ as } j \to \infty \tag{3.53}$$

and

$$\lim_{j\to\infty}||m_{2n_j+1} - p_{2n_j+1}|| = 0. \tag{3.54}$$

Therefore, by (3.50) - (3.54) and the metric projection property in (2.15), we find that

$$\limsup_{j\to\infty}\left\langle m_{2n_j} - z, u - z\right\rangle = \max_{x^*\in\omega_w(x_{2n})}\langle x^* - z, u - z\rangle \le 0$$

and

$$\limsup_{j\to\infty}\left\langle m_{2n_j+1} - z, u - z\right\rangle = \max_{x^*\in\omega_w(x_{2n})}\langle x^* - z, u - z\rangle \le 0. \tag{3.55}$$

Thus, by the condition $(A4)$, Assumption 2, Remark 3.2 and (3.55), we see that $\limsup_{j\to\infty}\Gamma_{2n_j} \le 0$. Therefore, it follows from Lemma 2.2 that $\lim_{n\to\infty}||x_{2n} - z^*|| = 0$ and hence $x_{2n} \to z^* = P_\Omega 0$ as $n \to \infty$.

Finally, combining the fact that $\lim_{n\to\infty}||x_{2n} - z^*|| = 0$ and (3.51), we see that $\lim_{n\to\infty}||x_{2n+1} - z^*|| = 0$. Thus we conclude that the odd subsequence $\{x_{2n+1}\}$ of $\{x_n\}$ produced by Algorithm 1 converges strongly to $z^* \in \Omega$. Hence the whole sequence $\{x_n\}$ produced by Algorithm 1 strongly converges to $z^* \in \Omega$. This completes the proof. □

To obtain some extensions of Algorithm 1, we make the following assumption.

Assumption 3: Let $k \in K_n \subseteq \{0, 1, 2, \cdots, n-1\}$, y_{n-k} and y_{n-k-1} be arbitrary points in $\mathcal{H}_1$ for all $n \ge 1$. Choose $\varsigma_{n,k}$, $\sigma_{n,k} \in [0, +\infty)$ such that $\lim_{n\to\infty}\frac{\sum_{k\in K_n}\varsigma_{n,k}}{\delta_n} = 0$ and $\lim_{n\to\infty}\frac{\sum_{k\in K_n}\sigma_{n,k}}{\delta_n} = 0$. Select $\beta_{n,k} \in [0, \bar\beta_{n,k}]$, $\delta_{n,k} \in [0, \bar\delta_{n,k}]$ for all $n \ge 1$, $k \in K_n$ and any $\eta_2, \eta_3 > 0$ such that

$$\bar\beta_{n,k} := \begin{cases} \min\left\{\frac{\varsigma_{n,k}}{||y_{n-k} - y_{n-k-1}||},\ \eta_2\right\}, & \text{if } y_{n-k} \ne y_{n-k-1}, \\ \eta_2, & \text{otherwise} \end{cases} \tag{3.56}$$

and

$$\bar{\delta}_{n,k} := \begin{cases} \min\left\{\frac{\sigma_{n,k}}{||y_{n-k}-y_{n-k-1}||},\ \eta_3\right\}, & \text{if } y_{n-k} \neq y_{n-k-1}, \\ \eta_3, & \text{otherwise.} \end{cases} \tag{3.57}$$

Remark 3.5. *We can easily see from Assumption 3 that, for every $n \geq 1$ and $k \in K_n$,*

$$\beta_{n,k}||y_{n-k} - y_{n-k-1}|| \leq \varsigma_{n,k}, \quad \delta_{n,k}||y_{n-k} - y_{n-k-1}|| \leq \sigma_{n,k}.$$

Then, in view of the fact that, for every $k \in K_n$,

$$\lim_{n\to\infty} \frac{\sum_{k\in K_n} \varsigma_{n,k}}{\delta_n} = 0, \quad \lim_{n\to\infty} \frac{\sum_{k\in K_n} \sigma_{n,k}}{\delta_n} = 0,$$

we, respectively, obtain that

$$\lim_{n\to\infty} \frac{\sum_{k\in K_n} \beta_{n,k}||y_{n-k} - y_{n-k-1}||}{\delta_n} = 0, \quad \lim_{n\to\infty} \frac{\sum_{k\in K_n} \delta_{n,k}||y_{n-k} - y_{n-k-1}||}{\delta_n} = 0. \tag{3.58}$$

So, it is not difficult to observe that, for each $n \geq 1$, taking

$$e_1(y_n) = \sum_{k\in K_n} \beta_{n,k}(y_{n-k} - y_{n-k-1}) \tag{3.59}$$

and

$$e_2(y_n) = \sum_{k\in K_n} \delta_{n,k}(y_{n-k} - y_{n-k-1}), \tag{3.60}$$

then Algorithm 1 becomes the following alternated and multi-step inertial Halpern-type relaxed algorithm for the problem (1.1).

Algorithm 2 Alternated and Multi-step Inertial Halpern-type Relaxed Algorithm (AMiHRA)

Initialization: Take $\tau_1,\ \varepsilon,\ \rho$ and $\{\delta_n\}$ such that the condition $(A4)$ of Assumption 1 holds. Select $K_n,\ \beta_{n,k}$ and $\delta_{n,k}$ for all $k \in K_n$ as described in Assumption 3, λ_n as in Remark 3.2, $u \in \mathcal{C}$, $y_0,\ x_0,\ x_1 \in \mathcal{H}_1$ and set $n = 1$.

Step 1. Compute y_n by (3.18).

Step 2. Compute

$$w_n = y_n + \sum_{k\in K_n} \beta_{n,k}(y_{n-k} - y_{n-k-1})$$

and

$$h_n = P_{\mathcal{C}_n}(w_n - \rho\tau_n\nabla g_n(y_n)).$$

If $h_n = w_n = y_n$, then stop the iteration and $h_n \in \Omega$. Else, go to Step 3.

Step 3. Compute

$$u_n = y_n + \sum_{k\in K_n} \delta_{n,k}(y_{n-k} - y_{n-k-1})$$

and

$$x_{n+1} = \delta_n u + (1-\delta_n)P_{\mathcal{C}_n}(u_n - \rho\tau_n\nabla g_n(h_n)),$$

update the step-length τ_{n+1} by (3.19), set $n := n+1$ and go back to Step 1.

Theorem 3.2. *Let $\{x_n\}$ be a sequence produced by Algorithm 2 such that the conditions of Assumption 1, 3 and Remark 3.2 hold. Then the sequence $\{x_n\}$ strongly converges to a point $z^* \in \Omega$, where $z^* = P_\Omega 0$.*

Proof. In view of the choice of $\varsigma_{n,k},\ \sigma_{n,k},\ \beta_{n,k},\ \delta_{n,k}$ for all $n \geq 1$ and $k \in K_n$ in Assumption 3 and the equations (3.56), (3.57), (3.59) and (3.60), it is clear that the conditions of Assumption 2 are satisfied when Assumption 3 holds. Therefore, the complete proof of Theorem 3.2 follows from that of Theorem 3.1. This completes the proof. □

Remark 3.6. *in the following remarks, we consider some new and existing algorithms for solving the problem (1.1) related to Algorithm 2:*

(1) If $K_n = \{0\}, \varsigma_{n,k} = \varsigma_n,\ \sigma_{n,k} = \sigma_n,\ \beta_{n,k} = \beta_n$ and $\delta_{n,k} = \delta_n$ for all $n \geq 1$ in Assumption 3, then the AMiHRA becomes a relaxed $\mathcal{CQ}$ algorithm that combines an alternated inertial step and two classical Polyak's inertial steps.

(2) If $K_n = \{0\}$ and $\beta_{n,k} = \delta_{n,k} = 0$ for all $n \geq 1$, then the AMiHRA reduces to Halpern-type of Algorithm 3.1 in [53] with monotonic step-length criterion.

(3) If $\lambda_n = 0$ for all $n \geq 1$, then the AMiHRA becomes a general multi-step inertial Halpern-type relaxed $\mathcal{CQ}$ algorithm that combines two multi-step inertial terms for the problem (1.1).

We also consider the following as another extension of Algorithm 1:

Algorithm 3 Accelerated Alternated and Multi-step Inertial Halpern-type Relaxed Algorithm (AAMiHRA)

Initialization: Take $\tau_1,\ \varepsilon,\ \rho$ and $\{\delta_n\}$ such that the condition $(A4)$ holds. Select $\sigma > 0,\ \omega_n,\ \varsigma_n^{(2)} \in [0, +\infty)$ such that $\lim\limits_{n\to\infty} \frac{\omega_n}{\delta_n} = 0,\ \lim\limits_{n\to\infty} \frac{\varsigma_n^{(2)}}{\delta_n} = 0$, K_n, $\delta_{n,k}$ for all $k \in K_n$ as described in Assumption 3, λ_n as in Remark 3.2 and a bounded sequence $\{s_n\} \subset \mathcal{H}_1$. Choose $u \in \mathcal{C}$, $y_0,\ x_0,\ x_1 \in \mathcal{H}_1$ and set $n = 1$.

Step 1. Compute y_n by (3.18).

Step 2. Compute

(3.61)
$$\varsigma_n^{(1)} = \frac{\omega_n}{\max\{||d_n||, \sigma\}},$$

(3.62)
$$d_{n+1} = \begin{cases} -\nabla g_n(y_n), & \text{if } n = 0, \\ -\frac{1}{\vartheta}\rho\tau_n \nabla g_n(y_n) + \varsigma_n^{(1)} d_n - \varsigma_n^{(2)} s_n, & \text{otherwise} \end{cases}$$

and

$$h_n = P_{\mathcal{C}_n}(y_n + \vartheta d_{n+1}).$$

If $h_n = y_n$, then stop the iteration and $h_n \in \Omega$. Else, go to Step 3.

Step 3. Compute

$$u_n = y_n + \sum_{k\in K_n} \delta_{n,k}(y_{n-k} - y_{n-k-1}),$$

and

$$x_{n+1} = \delta_n u + (1 - \delta_n) P_{\mathcal{C}_n}(u_n - \rho\tau_n \nabla g_n(h_n)),$$

update the step-length τ_{n+1} by (3.19), set $n := n + 1$ and go back to Step 1.

Remark 3.7. *Observe that taking $e_2(y_n)$ as in (3.60) and defining $e_1(y_n)$ as follows:*

(3.63)
$$e_1(y_n) = \vartheta\big(\varsigma_n^{(1)} d_n - \varsigma_n^{(2)} s_n\big),$$

with $\varsigma_n^{(1)}$ for all $n \geq 1$ to be obtained by (3.61), then, from the conditions on $\omega_n,\ \varsigma_n^{(2)}$ and the boundedness of the sequence $\{s_n\}$, Algorithm 1 becomes Algorithm 3. Thus we formulate and prove the following theorem:

Theorem 3.3. *Suppose that the conditions of Assumption 1 and 3 hold, $\{e_1(y_n)\}$ and $\{e_2(y_n)\}$ are the sequences generated by (3.63) and (3.60), respectively, $\{s_n\} \subset \mathcal{H}_1$ is bounded and $\{x_n\}$ is a sequence produced by Algorithm 3. Then $\{x_n\}$ strongly converges to a point $z^* \in \Omega$, where $z^* = P_\Omega 0$.*

Proof. In view of the choice of ω_n, $\varsigma_n^{(2)}$ in Algorithm 3, $\sigma_{n,k}$ and $\delta_{n,k}$ for all $n \geq 1$ and $k \in K_n$ in Assumption 3, the boundedness of $\{s_n\}$ and the equations (3.57), (3.60) and (3.63), it is obvious that the conditions of Assumption 2 are satisfied when Assumption 3 and Remark 3.7 hold. Therefore, the complete proof of the Theorem 3.3 follows from that of Theorem 3.1. This completes the proof. □

Remark 3.8. *We provide some new brand of self-adaptive relaxed $\mathcal{CQ}$ Algorithms for solving the problem (1.1) based on Algorithm 3.*

(1) If $K_n = \{0\}$ and $\delta_{n,k} = \delta_n$ for all $n \geq 0$ in Assumption 3, then the AAMiHRA becomes a general accelerated inertial Halpern-type relaxed algorithm, which combines an alternated inertial step, the classical Polyak's inertial step and a three-term conjugate-like direction in a single algorithm with monotonically decreasing step-length criterion.

(2) If $K_n = \{0\}$ and $\delta_{n,k} = \delta_n = 0$ for all $n \geq 1$, then the AAMiHRA reduces a Halpern-type of the alternated inertial algorithm 3.1 in [53] with three-term conjugate gradient-like direction and monotonic step-length criterion.

(3) If $\varsigma_n^{(i)} = 0$ for all $i = 1, 2$ and $n \geq 1$, then the AAMiHRA reduces an alternated and multi-step inertial Halpern-type relaxed algorithm with monotonic step-length criterion.

(4) If $K_n = \{0\}$, $\delta_{n,k} = \delta_n = 0$ and $\varsigma_n^{(i)} = 0$ for all $i = 1, 2$ and $n \geq 1$, then the AAMiHRA reduces a Halpern-type of the alternated inertial algorithm 3.1 in [53] with monotonic step-length criterion.

4. Numerical Experiments

In this section, we investigate the performance and efficiency of the proposed algorithms (i.e., AMiHRA and AAMiHRA) in solving classification problems and constrained minimization problems. We conducted the experiments using R2023a Matlab in a PC with 12th Gen Intel(R) Core(TM)i5-124P 1.70 GHz processor and 16.0GB RAM.

4.1. The Constrained Minimization Problem.

In this part, we consider the following constrained minimization problem:

$$\min_{x \in \mathcal{C}} \frac{1}{2} ||\mathcal{B}x - P_\mathcal{Q}\mathcal{B}x||^2, \tag{4.64}$$

where $\mathcal{C} = \{x \in L_2[0,1] : \langle x(t), 3t^2 \rangle = 0\}$ and $\mathcal{Q} = \{x \in L_2[0,1] : \langle x(t), \frac{t}{3} \rangle \geq -1\}$ are in $L_2[0,1]$.

Setting $g(x) = \frac{1}{2}||\mathcal{B}x - P_\mathcal{Q}\mathcal{B}x||^2$, it is not difficult to see that $\nabla g(x) = \mathcal{B}^*(I - P_\mathcal{Q})\mathcal{B}x$ is $||\mathcal{B}||^2$-Lipschitch continuous. Thus, problem (4.64) can be transformed into problem (1.1) with $\mathcal{H}_1 = \mathcal{H}_2 = L_2[0,1]$, where $||x|| = \left(\int_0^1 |x(t)|^2 dt\right)^{1/2}$ and $\langle x, y \rangle = \int_0^1 x(t)y(t)dt$ are, respectively, the norm and the inner product in $L_2[0,1]$. For all the experiments, we consider $\mathcal{B} = I$, where I is the identity mapping, i.e., $\mathcal{B}x = x$. Since $\mathcal{Q}$ and $\mathcal{C}$ are half-space and hyper-plane, respectively, to apply our proposed algorithms (i.e., AMiHRA and AAMiHRA) to solve problem (4.64), we take $\mathcal{Q}_n = \mathcal{Q}$ for all $n \geq 1$, define $c(x) = \langle x(t), 3t^2 \rangle$ for all $x \in L_2[0,1]$, so that $\mathcal{C}$ satisfies (1.5) and we consider g_n and $\mathcal{C}_n$ as described in (1.3) and (1.6), respectively, and set $\nabla g_n = \mathcal{B}^*(I - P_{\mathcal{Q}_n})\mathcal{B}$. We use the defined explicit projection formula in [24] to compute the projection

$P_{\mathcal{Q}_n}$ and the projection $P_{\mathcal{C}_n}$ by the following:

$$P_{\mathcal{C}_n}(x_n(t)) = \begin{cases} x_n(t), \text{ if } \langle 3t^2, t - x_n(t)\rangle \geq c(x_n(t)), \\ x_n(t) - \frac{c(x_n(t))+\langle 3t^2, x_n(t)-t\rangle}{||3t^2||^2_{L_2}} 3t^2, \text{ otherwise}. \end{cases} \tag{4.65}$$

We compare the performance of our algorithms, the AMiHRA and the AAMiHRA with the algorithms of Tan et al. [53] and Dong et al. [21], which we abbreviated in this work as TQW Alg 3.1 and DLY Alg 4-II, respectively. For the experiments, we select the following parameters:

(1) We set $\tau_1 = 0.058,\ \eta_1 = \eta_3 = 5,\ \varepsilon = 0.1,\ \rho = \frac{1}{\varepsilon} - \frac{1}{3000\varepsilon}, \delta_n = \frac{1}{10^5 n+1},\ \xi_n = \frac{1}{(n+10)^2}$, and $\sigma_{n,k} = \frac{1}{n^5 k^5}$ for the AMiHRA and the AAMiHRA. In particular, we select $\eta_2 = 5$ and $\varsigma_{n,k} = \frac{1}{n^3 k^3}$ for the AMiHRA and $\sigma = 0.1,\ \vartheta = 3,\ \omega_n = \frac{1}{(n+10)^5}$ and $\varsigma_n^{(2)} = \frac{1}{(10n+1)^3}$ for the AAMiHRA.
(2) In TQW Alg 3.1, we choose $\lambda_1 = 0.058,\ \mu = 0.1,\ \beta = 1.3,\ \alpha = 1,\ \theta_n = 0.2,\ \rho_n = \frac{10^{-1}}{(n+1)^2}$ and $\xi_n = 1 + \frac{10^{-1}}{(n+1)^2}$.
(3) In DLY Alg 4-II, we set $\tau_1 = 0.058,\ \varepsilon = 0.1,\ \rho = 0.2$ and $\lambda_n = \frac{1}{50n+1} - 1$.

For the implementations of the algorithms, we consider four different cases of the initial values of $x_0,\ x_1,\ y_0, u$ and s_n:
Case I: $x_0(t) = \sin(t),\ x_1(t) = t^2,\ y_0(t) = 0.5t^2,\ u(t) = t$ and $s_n(t) = 1.7t$;
Case II: $x_0(t) = e^t,\ x_1(t) = t^3,\ y_0(t) = 3\sin(t),\ u(t) = t$ and $s_n(t) = 10\sqrt{t}$;
Case III: $x_0(t) = \cos(t),\ x_1(t) = \tanh t,\ y_0(t) = 0.5t^2,\ u(t) = \frac{t}{100}$ and $s_n(t) = 5t^3$;
Case IV: $x_0(t) = e^{3t^2},\ x_1(t) = t^5,\ y_0(t) = \sin t^2,\ u(t) = \frac{\sqrt[3]{t}}{10}$ and $s_n(t) = \sqrt[4]{t}$.
We used the stopping rule

$$E_n = \frac{1}{2}\big(||x_n(t) - P_{\mathcal{C}_n} x_n(t)||^2_{L_2} + ||x_n(t) - P_{\mathcal{Q}_n} x_n(t)||^2_{L_2}\big) < 10^{-10}$$

and the maximum number of iterations of 200 to terminate the iterations for all the algorithms. The performance results of all the algorithms, which include the execution times in second represented by "Time", the number of iterations denoted by "Iter." and the error E_n are reported in Table 1 and we plot the corresponding error results for the four cases in Figures 1, 2, 3 and 4.

TABLE 1. Compare the performance of the algorithms for the four cases

Cases	Algorithms	Iter.	Time(s)	E_n
Case I	AMiHRA	30	0.0531	4.98E-14
	AAMiHRA	30	0.0507	1.17E-13
	TQW Alg 3.1	98	0.0827	7.94E-11
	DLY Alg 4-II	115	0.0926	8.94E-11
Case II	AMiHRA	30	0.0482	2.64E-13
	AAMiHRA	30	0.0415	2.70E-12
	TQW Alg 3.1	98	0.0731	7.70E-11
	DLY Alg 4-II	115	0.0925	8.67E-11
Case III	AMiHRA	30	0.0439	2.45E-13
	AAMiHRA	30	0.0439	6.13E-14
	TQW Alg 3.1	98	0.0789	8.10E-11
	DLY Alg 4-II	115	0.0828	9.11E-11
Case IV	AMiHRA	30	0.0443	1.43E-13
	AAMiHRA	30	0.0389	2.10E-13
	TQW Alg 3.1	98	0.0733	7.43E-11
	DLY Alg 4-II	115	0.0903	8.36E-11

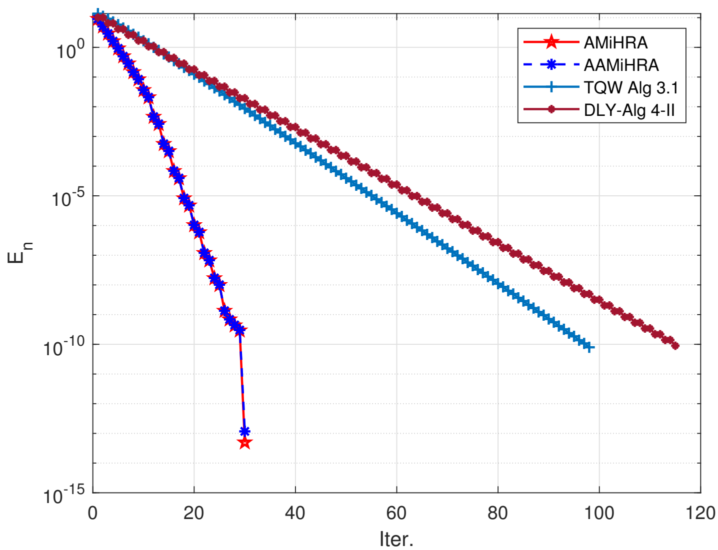

FIGURE 1. Error plotting of E_n of all the algorithms for Case I.

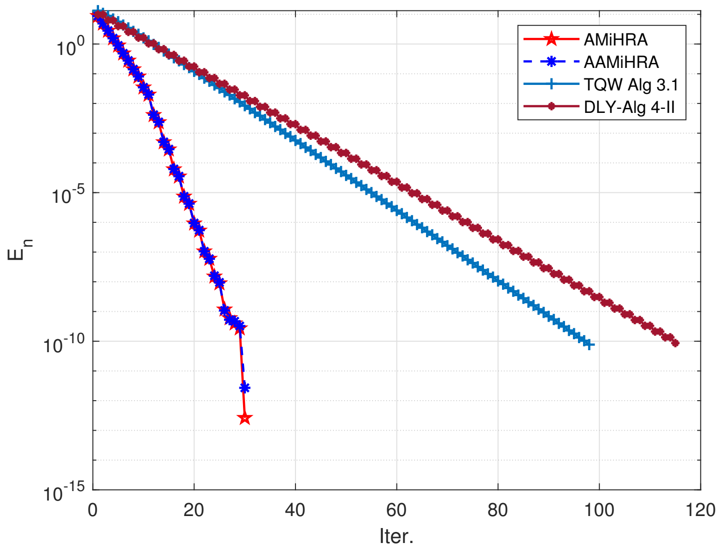

FIGURE 2. Error plotting of E_n of all the algorithms for Case II.

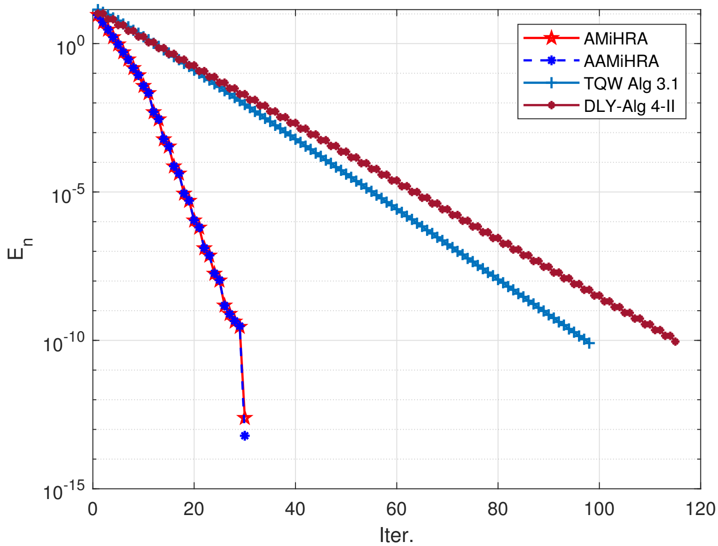

FIGURE 3. Error plotting of E_n of all the algorithms for Case III.

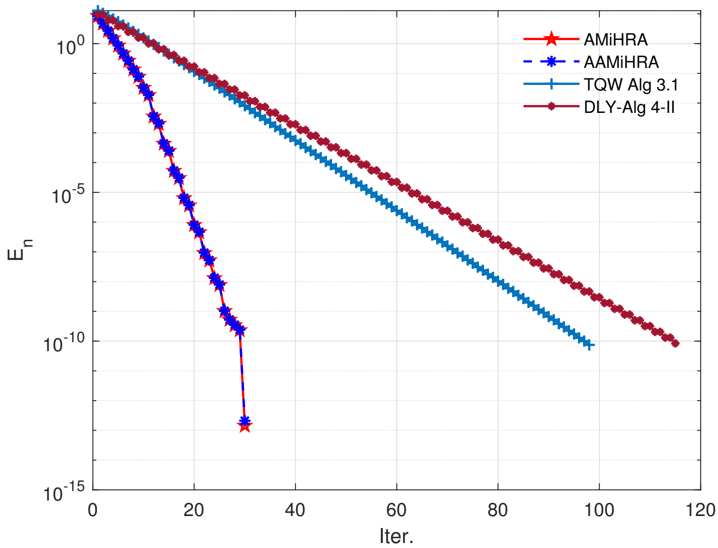

FIGURE 4. Error plotting of E_n of all the algorithms for Case IV.

Remark 4.9. *We observed from Table 1 and Figures 1, 2, 3 and 4 that the proposed algorithms outperform the compared algorithms in all the experiments. In particular, the AMiHRA achieves the fewest errors in most of the experiments, while AAMiHRA has the shortest execution times in all the experiments.*

4.2. **Classification Problems.** In this part, we conduct a series of experiments on some real-world benchmark datasets to investigate the performance of the suggested algorithms (i.e., AMiHRA and AAMiHRA). In all the experiments, we consider an efficient learning algorithm called extreme learning machine ELM for single-hidden layer feedforward neural networks SLFNs [29] and take $\mathcal{K} = \{(x_j, t_j) \in \mathbb{R}^k \times \mathbb{R}^m,\ j = 1, 2, \cdots, \mathcal{N}\}$ as an $\mathcal{N}$ distinct training data points set, where for each input point $x_j = \left[x_{j1}, x_{j2}, \cdots, x_{jk}\right]^T$, $t_j = \left[t_{j1}, t_{j2}, \cdots, t_{jm}\right]^T$ is its corresponding target. The SLFNs output function with $\mathcal{L}$ number of nodes in the hidden layer has the following formulation.

$$g_j = \sum_{i=1}^{\mathcal{L}} \beta_i f_i(x_j), \quad \forall j = 1, 2, \cdots, \mathcal{N}, \tag{4.66}$$

where $f_i(x_j) = \mathcal{F}\Big(\langle \omega_i, x_j \rangle + b_i\Big)$, $\mathcal{F}$ is an activation function, $\omega_i = (\omega_{i1}, \omega_{i2} \cdots, \omega_{ik})^T$ is an input weight vector linking the i^{th} hidden node and the input nodes, $\beta_i = (\beta_{i1}, \beta_{i2}, \cdots, \beta_{im})^T$ is an output weight vector linking the i^{th} hidden node and the output nodes and b_i is a bias. To train a SLFNs is to solve the linear system:

$$\mathcal{G}\beta = \mathsf{T}, \tag{4.67}$$

where the hidden layer output matrix $\mathcal{G}$ of order $\mathcal{N} \times \mathcal{L}$ is given by

$$\mathcal{G} = \left[f_1(x), f_2(x), \cdots, f_{\mathcal{L}}(x)\right],$$

$\beta = (\beta_1, \beta_2, \cdots, \beta_{\mathcal{L}})^T$ and $\mathsf{T} = (t_1, t_2, \cdots, t_{\mathcal{N}})^T$ are the output weights and the target data matrices, respectively and the i^{th} column of $\mathcal{G}$ is the i^{th} hidden node output based on $x_1, x_2 \cdots, x_{\mathcal{N}}$, which is defined by $f_i(x) = \left[f_i(x_1), f_i(x_2), \cdots, f_i(x_{\mathcal{N}})\right]^T$. To solve (4.67) by ELM is simply to find an optimal output weight $\hat{\beta} = \mathcal{G}^\dagger \mathsf{T}$, where $\mathcal{G}^\dagger$ represents the Moore-Penrose generalized inverse of the matrix $\mathcal{G}$ [47].

From the perspective of the sparsity of the output weight parameter β for some high-dimensional data, Cao et al. [8] proposed an ℓ_1-regularization approach to solve problem (4.67) based on the following Lasso model [54]:

$$\min_{\beta \in \mathbb{R}^{\mathcal{L}\times m}} \left\{ \frac{1}{2}||\mathsf{T} - \mathcal{G}\beta||_2^2 \ : \ ||\beta||_1 \leq c \right\}, \tag{4.68}$$

where $c > 0$ is the regularization parameter. However, for better prediction accuracy, sparsity and stability, Ye et al. [61] unified both the ℓ_1 and the ℓ_2 penalties into a single model called the $\ell_1 - \ell_2$ hybrid regularization approach. Their model is described as follows.

$$\min_{\beta \in \mathbb{R}^{\mathcal{L}\times m}} \left\{ \frac{1}{2}||\mathsf{T} - \mathcal{G}\beta||_2^2 \ : \ \lambda||s||_1 + \gamma||s||_2^2 \leq c \right\}, \tag{4.69}$$

where $\lambda, \ \gamma \geq 0$ and $c > 0$ are the regularization parameters. Suantai et al. [50] transformed the problem (4.68) into problem (1.1) by taking $\mathcal{C} = \{\beta \in \mathbb{R}^{\mathcal{L}\times m} \ : \ ||\beta||_1 \leq c\}$, $\mathcal{Q} = \{\mathsf{T}\} \subseteq \mathbb{R}^{\mathcal{K}\times m}$, $c(\beta) = ||\beta||_1 - c$, $q(x) = \frac{1}{2}||x - \mathsf{T}||^2$ and defined g_n, $\mathcal{C}_n$ and $\mathcal{Q}_n$ as in (1.3) and (1.6), respectively. They also used their proposed inertial relaxed $\mathcal{CQ}$ algorithm to solve the problem (4.67) based on the model (4.68).

Inspired by the sparsity, the stability and the generalization performance of (4.69), we observe that transforming problem (4.69) into problem (1.1) is of paramount important, which is possible by taking

$$\mathcal{C} = \{\beta \in \mathbb{R}^{\mathcal{L}\times m} \ : \ \lambda||\beta||_1 + \gamma||\beta||_2^2 \leq c\},$$
$$\mathcal{Q} = \{\mathsf{T}\} \subseteq \mathbb{R}^{\mathcal{K}\times m}, \quad q(x) = \frac{1}{2}||x - \mathsf{T}||^2.$$

Moreover, it is easily seen that the function $c(\beta) = \lambda||\beta||_1 + \gamma||\beta||_2^2 - c$ is strongly convex, so it is convex. We consider g_n, $\mathcal{C}_n$ and $\mathcal{Q}_n$ as defined in (1.3) and (1.6), respectively. Therefore, our proposed algorithms (i.e., AMiHRA and AAMiHRA) can be used to solve the problem (4.67) based on the both models (4.68) and (4.69).

To investigate the performance of the proposed algorithms, we employed them to solve problem (4.67) based on the models (4.68) and (4.69), for which we used the abbreviations AMiHRA - ℓ_1, AMiHRA - $\ell_1 - \ell_2$, AAMiHRA - ℓ_1 and AAMiHRA - $\ell_1 - \ell_2$ to denote them respectively. We compare their results with the algorithms of Tan et al. [53], Dong et al. [21] and Abubakar et al. [1] based on the model (4.68), which we respectively abbreviated in this work as TQW Alg 3.1 $- \ell_1$, DLY Alg 4-II - ℓ_1 and AKTIS Alg 1 $- \ell_1$. We carried out the experiments on three real-world classification datasets, including Breast Cancer Wisconsin (Breast Cancer W.) dataset [55], Heart disease dataset [34] and Glass identification dataset [22]. The detailed information on each of the datasets is provided in Table 2.

TABLE 2. Details of each dataset

Datasets	Instances	Classes	Features	Tasks
Breast Cancer W.	569	2	30	Classification
Heart disease	303	2	13	Classification
Glass Identification	214	6	9	Classification

In all the experiments, we fixedly choose 70% of each of the datasets for training and 30% for testing. We also set the following for the parameters:

(1) We set $\tau_1 = 3.03 \times 10^{-5},\ \eta_1 = \eta_2 = \eta_3 = 3,\ \varepsilon = 0.012,\ \rho = 0.13,\ \delta_n = \frac{1}{10^5 n+1},\ \xi_n = \frac{1}{(n+10)^{3.4}}$, $\varsigma_{n,k} = \frac{1}{n^3 k^3}$ and $\sigma_{n,k} = \frac{1}{n^5 k^5}$ for AMiHRA - ℓ_1, AMiHRA - $\ell_1 - \ell_2$, AAMiHRA - ℓ_1 and AAMiHRA - $\ell_1 - \ell_2$. In particular, we select $\sigma = 0.1,\ \vartheta = 3,\ \omega_n = \frac{1}{(n+10)^5}$ and $\varsigma_n^{(2)} = \frac{1}{(10n+1)^3}$ for AAMiHRA - ℓ_1 and AAMiHRA - $\ell_1 - \ell_2$.

(2) In TQW Alg $3.1 - \ell_1$, we choose $\lambda_1 = 3.03 \times 10^{-5},\ \mu = 0.012,\ \beta = 0.13,\ \alpha = 0.997,\ \theta_n = 0.002,\ \rho_n = \frac{10^{-1}}{(n+1)^2}$ and $\xi_n = 1 + \frac{10^{-1}}{(n+1)^2}$.

(3) In DLY Alg 4-II - ℓ_1, we set $\tau_1 = 3.03 \times 10^{-5},\ \varepsilon = 0.012,\ \rho = 0.13$ and $\lambda_n = \frac{1}{50n+1} - 1$.

(4) In AKTIS Alg $1 - \ell_1$, we select $\varrho = 0.5,\ \varepsilon_n = \frac{1}{n^2},\ \zeta_n = \frac{1}{7500(n+5)}$ and $\vartheta_n = 0.8 - \zeta_n$.

We respectively calculate the accuracies and precisions by the following relations.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \times 100\%, \tag{4.70}$$

$$\text{Precision} = \frac{\text{TP}}{\text{FP} + \text{FN}} \times 100\%, \tag{4.71}$$

where TP := True positive, TN := True negative, FP = False positive and FN = False negative, and estimate their averages as well as their standard deviations (SDs). We use these metrics and the number of iterations denoted by "Iter." to investigate the effectiveness and the stability of the suggested algorithms.

In the first part of the experiments, we set $eC := \text{ones}(\mathcal{L}, m),\ x_0 = -1eC,\ x_1 = eC,\ u = y_0 = 2eC,\ s_n = 1.7eC$, $\mathcal{F}(x) = \tanh(x)$ as the activation function, $c = 0.061,\ \lambda = 0.9999,\ \gamma = 0.00505$ and used $||x_{n+1} - x_n|| < 10^{-3}$ and 500 as the Maximum iteration count to terminate the iterations for all the algorithms. We then analyzed the sensitivity of all the algorithms on the Breast Cancer W. dataset over different number of hidden nodes. The performance of all the algorithms are shown in Table 3 and we plot the corresponding results on training and testing accuracies, and training and testing precisions in Figures 5, 6, 7 and 8, respectively.

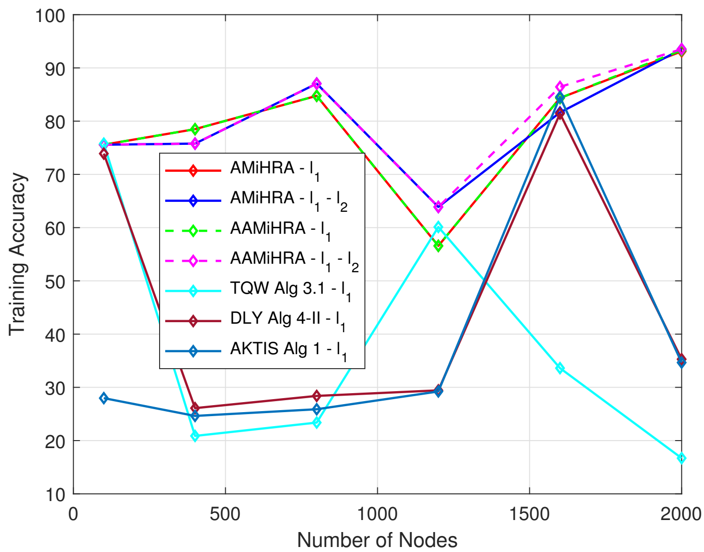

FIGURE 5. Compare the training accuracies of all the algorithms over different number of hidden nodes on the Breast Cancer W. dataset using the activation function $\mathcal{F}(x) = \tanh(x)$.

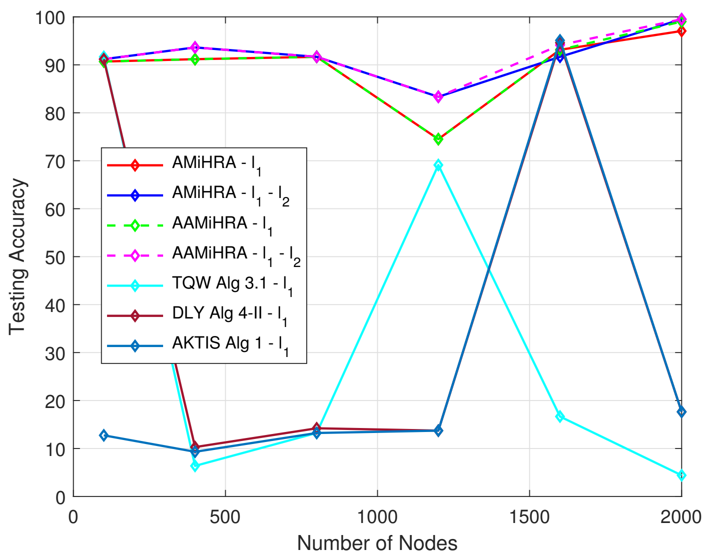

FIGURE 6. Compare the testing accuracies of all the algorithms over different number of hidden nodes on the Breast Cancer W. dataset using the activation function $\mathcal{F}(x) = \tanh(x)$.

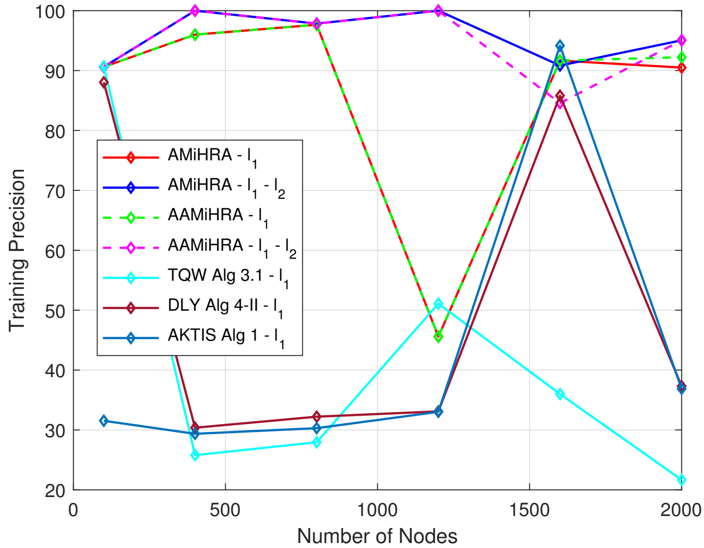

FIGURE 7. Compare the training precisions of all the algorithms over different number of hidden nodes on the Breast Cancer W. dataset using the activation function $\mathcal{F}(x) = \tanh(x)$.

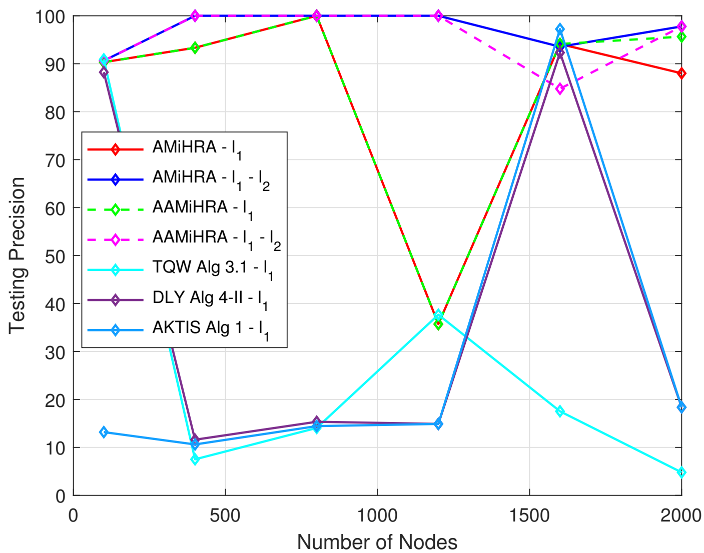

FIGURE 8. Compare the testing precisions of all the algorithms over different number of hidden nodes on the Breast Cancer W. dataset using the activation function $\mathcal{F}(x) = \tanh(x)$.

TABLE 3. Performance results of all the algorithms on the Breast Cancer W. dataset. The best and suboptimal results are highlighted in bold and underlined, respectively.

Accuracy (%)

Algorithms	AMiHRA - ℓ_1		AMiHRA - $\ell\_1 - \ell_2$		AAMiHRA - ℓ_1		AAMiHRA - $\ell_1 - \ell_2$		TQW Alg $3.1 - \ell_1$		DLY Alg 4-II - ℓ_1		AKTIS Alg $1 - \ell_1$	
Nodes	Training	Testing	Training	Testing	Training	Testing	Training	Testing	Training	Testing	Training	Testing	Training	Testing
100	75.5741	90.6863	75.5741	91.1765	75.5741	90.6863	75.5741	91.1765	**75.7829**	**91.6667**	73.904	91.1765	27.9749	12.7451
400	**78.4969**	91.1765	75.7829	**93.6275**	**78.4969**	91.1765	75.7829	**93.6275**	20.8768	6.3725	26.096	10.2941	24.6347	9.3137
800	84.7599	**91.6667**	**87.0564**	**91.6667**	84.7599	**91.6667**	**87.0564**	**91.6667**	23.382	13.2353	28.3925	14.2157	25.8873	13.2353
1200	56.5762	74.5098	**63.8831**	**83.3333**	56.5762	74.5098	**63.8831**	**83.3333**	60.1253	69.1176	29.4363	13.7255	29.2276	13.7255
1600	84.3424	93.1373	81.6284	91.6667	84.3424	93.1373	**86.4301**	94.1176	33.6117	16.6667	81.4196	94.6078	84.5511	**95.098**
2000	93.1106	97.0588	**93.5282**	**99.5098**	93.3194	99.0196	**93.5282**	**99.5098**	16.7015	4.4118	35.2818	17.6471	34.6555	17.6471
Aver. Acc.	78.81	89.7059	79.5755	91.8301	78.8448	90.0327	**80.3758**	**92.2386**	38.4134	33.5784	45.755	40.2778	37.8219	26.9608
SDs	12.4533	7.794	**10.3102**	**5.1939**	12.5015	8.1947	10.6811	5.2743	24.0658	37.2236	25.013	40.8359	23.1552	33.4858

Precision (%)

Algorithms	AMiHRA - ℓ_1		AMiHRA - $\ell_1 - \ell_2$		AAMiHRA - ℓ_1		AAMiHRA - $\ell_1 - \ell_2$		TQW Alg $3.1 - \ell_1$		DLY Alg 4-II - ℓ_1		AKTIS Alg $1 - \ell_1$	
Nodes	Training	Testing	Training	Testing	Training	Testing	Training	Testing	Training	Testing	Training	Testing	Training	Testing
100	90.625	90.3226	90.625	90.625	90.625	90.3226	90.625	90.625	**90.7216**	**90.9091**	88.0435	88.2353	31.5271	13.1868
400	96.00	93.3333	**100**	**100**	96.00	93.3333	**100**	**100**	25.7895	7.5145	30.3704	11.6022	29.3532	10.6145
800	97.6562	**100**	**97.8417**	**100**	97.6562	**100**	**97.8417**	**100**	27.9487	14.0541	32.2115	15.3439	30.2956	14.4385
1200	45.5782	35.7143	**100**	**100**	45.5782	35.7143	**100**	**100**	51.087	37.6623	33.0969	14.8936	33.0189	14.8936
1600	91.6667	94.1176	90.8397	93.5484	91.6667	94.1176	84.5745	84.7826	35.9909	17.5258	85.8108	92.3077	**94.1606**	**97.2222**
2000	90.50	88.00	**95.0549**	**97.7778**	92.228	95.6522	**95.0549**	**97.7778**	21.6667	4.7904	37.3068	18.3673	36.8889	18.3673
Aver. Prec.	85.3377	83.5813	**95.7269**	**96.9919**	85.6257	84.8567	94.6827	95.5309	42.2007	28.7427	51.14	40.125	42.5407	28.1205
SDs	19.7023	23.7988	**4.2745**	**4.004**	19.8052	24.2831	6.0878	6.3963	25.9457	32.5888	27.8227	38.9238	25.4259	33.9464

Remark 4.10. *Comparing the performance results of all the algorithms shown in Table 3 and Figures 5, 6, 7 and 8, we make the following remarks.*

(1) It is easily seen that our proposed algorithms, the AMiHRA - ℓ_1 and the AAMiHRA - ℓ_1 comparatively achieve higher training and testing accuracies and precisions than TQW Alg $3.1-\ell_1$, DLY Alg 4-II - ℓ_1 and AKTIS Alg $1-\ell_1$. Meanwhile, the SDs of both the training and testing accuracies and precisions of the AMiHRA - ℓ_1 and the AAMiHRA - ℓ_1 are extremely smaller than those of TQW Alg $3.1-\ell_1$, DLY Alg 4-II - ℓ_1 and AKTIS Alg $1-\ell_1$. These illustrate that the AMiHRA - ℓ_1 and the AAMiHRA - ℓ_1 achieve better stability and generalization performance in the experiments.

(2) It is also noted that due to the presence of the ℓ_2 penalty, the AMiHRA - $\ell_1-\ell_2$ and the AAMiHRA - $\ell_1-\ell_2$ have higher training and testing accuracies and precisions in most of the results than their corresponding AMiHRA - ℓ_1 and AAMiHRA - ℓ_1, which demonstrate their ability to achieve better generalization performance. Additionally, the SDs of both the training and testing accuracies and precisions of the AMiHRA - $\ell_1-\ell_2$ and the AAMiHRA - $\ell_1-\ell_2$ are extremely smaller than those of the AMiHRA - ℓ_1 and the AAMiHRA - ℓ_1, which show that they are more stable.

Though the effectiveness and stability of our proposed algorithms have been demonstrated in the aforementioned experiments, to further investigate their comparative performance in this practical applications, we still need to conduct more statistical analysis. In this regard, we used the three UCI datasets mentioned in our earlier discussion and four different activation functions to measure and compare the statistical performance of all the algorithms. In the second series of experiments, we set $eC := \text{ones}(\mathcal{L}, m), eQ := \text{randn}(\mathcal{L}, m)$, $x_0 = -1eQ,\ x_1 = eQ,\ y_0 = 2eQ,\ u = 10^{-5}eC,\ s_n = 1.7eC$. We choose $\mathcal{L} = 100$, and used $||x_{n+1} - x_n|| < 10^{-5}$ and 100 as the Maximum number of iterations to terminate the the process for all the algorithms. As depicted in Table 4, we set the parameters $c,\ \lambda$ and γ according to the dataset and the activation function. The training and testing accuracies as well as the number of iterations of all the algorithms are reported in Table 4. We further display the performance comparison results among the algorithms based on the number of wins, ties and looses in Tables 5.

Activation functions		Sigmoid				Radbas			
				Accuracy (%)				Accuracy (%)	
Datasets	Algorithms	$c,\ \lambda,\ \gamma$	Iter.	Training	Testing	$c,\ \lambda,\ \gamma$	Iter.	Training	Testing
	AMiHRA - ℓ_1	0.95	9	97.7011	**98.4615**	1.1	8	93.1034	90.7692
	AMiHRA - $\ell_1-\ell_2$	0.95, 0.9999, 0.00505	9	**98.2759**	**98.4615**	1.1, 0.9999, 0.00505	9	**94.8276**	**95.3846**
	AAMiHRA - ℓ_1	0.95	9	97.7011	**98.4615**	1.1	8	93.1034	90.7692
Breast Cancer W.	AAMiHRA - $\ell_1-\ell_2$	0.95, 0.9999, 0.00505	9	**98.2759**	**98.4615**	1.1, 0.9999, 0.00505	9	**94.8276**	**95.3846**
	TQW Alg 3.1 - ℓ_1	0.95	9	94.2529	95.3846	1.1	8	88.5057	80.00
	DLY Alg 4-II - ℓ_1	0.95	11	51.7241	58.4615	1.1	11	78.7356	78.4615
	AKTIS Alg 1 - ℓ_1	0.95	17	54.5977	63.0769	1.1	16	29.8851	26.1538
	AMiHRA - ℓ_1	2.7	7	**96.4912**	**93.4783**	2.7	7	**95.614**	**86.9565**
	AMiHRA - $\ell_1-\ell_2$	2.7, 0.999, 0.001	7	95.614	**93.4783**	2.7, 0.999, 0.001	7	**95.614**	**86.9565**
	AAMiHRA - ℓ_1	2.7	7	**96.4912**	**93.4783**	2.7	7	**95.614**	**86.9565**
Heart Disease	AAMiHRA - $\ell_1-\ell_2$	2.7, 0.999, 0.001	7	**96.4912**	**93.4783**	2.7, 0.999, 0.001	7	**95.614**	**86.9565**
	TQW Alg 3.1 - ℓ_1	2.7	7	33.3333	21.7391	2.7	7	42.9825	45.6522
	DLY Alg 4-II - ℓ_1	2.7	7	92.1053	91.3043	2.7	7	91.2281	**86.9565**
	AKTIS Alg 1 - ℓ_1	2.7	16	33.3333	30.4348	2.7	16	25.4386	26.087
	AMiHRA - ℓ_1	0.91	9	90.00	90.00	0.701	10	96.00	**100.00**
	AMiHRA - $\ell_1-\ell_2$	0.91, 0.999, 0.002	9	**98.00**	**95.00**	0.701, 0.99, 0.0107	9	**98.00**	**100.00**
	AAMiHRA - ℓ_1	0.91	9	90.00	90.00	0.701	10	96.00	**100.00**
Glass Identification	AAMiHRA - $\ell_1-\ell_2$	0.91, 0.999, 0.002	9	**98.00**	**95.00**	0.701, 0.99, 0.0107	9	**98.00**	**100.00**
	TQW Alg 3.1 - ℓ_1	0.91	9	42.00	35.00	0.701	10	68.00	90.00
	DLY Alg 4-II - ℓ_1	0.91	9	24.00	20.00	0.701	7	0.00	0.00
	AKTIS Alg 1 - ℓ_1	0.91	13	30.00	45.00	0.701	13	30.00	15.00

Activation functions		Tribas				Hardlim			
				Accuracy (%)				Accuracy (%)	
Datasets	Algorithms	$c,\ \lambda,\ \gamma$	Iter.	Training	Testing	$c,\ \lambda,\ \gamma$	Iter.	Training	Testing
	AMiHRA - ℓ_1	0.94	9	96.5517	95.3846	1.05	9	**96.5517**	96.9231
	AMiHRA - $\ell_1-\ell_2$	0.94, 0.9999, 0.00505	9	**98.2759**	**96.9231**	1.05, 0.999, 0.0009	9	95.977	98.4615
	AAMiHRA - ℓ_1	0.94	9	96.5517	95.3846	1.05	9	**96.5517**	96.9231
Breast Cancer W.	AAMiHRA - $\ell_1-\ell_2$	0.94, 0.9999, 0.00505	9	**98.2759**	**96.9231**	1.05, 0.999, 0.0009	9	95.977	**100.00**
	TQW Alg 3.1 - ℓ_1	0.94	9	63.2184	61.5385	1.05	9	92.5287	84.6154
	DLY Alg 4-II - ℓ_1	0.94	11	60.9195	53.8462	1.05	11	95.4023	98.4615
	AKTIS Alg 1 - ℓ_1	0.94	40	47.7011	40	1.05	16	58.046	61.5385
	AMiHRA - ℓ_1	2.7	7	**85.9649**	**80.4348**	2.7	7	**85.9649**	**80.4348**
	AMiHRA - $\ell_1-\ell_2$	2.7, 0.999, 0.001	7	**85.9649**	**80.4348**	2.7, 0.999, 0.001	7	**85.9649**	**80.4348**
	AAMiHRA - ℓ_1	2.7	7	**85.9649**	**80.4348**	2.7	7	**85.9649**	**80.4348**
Heart Disease	AAMiHRA - $\ell_1-\ell_2$	2.7, 0.999, 0.001	7	**85.9649**	**80.4348**	2.7, 0.999, 0.001	7	**85.9649**	**80.4348**
	TQW Alg 3.1 - ℓ_1	2.7	7	45.614	45.6522	2.7	7	30.7018	15.2174
	DLY Alg 4-II - ℓ_1	2.7	7	84.2105	76.087	2.7	7	77.193	67.3913
	AKTIS Alg 1 - ℓ_1	2.7	16	29.8246	28.2609	2.7	16	25.4386	21.7391
	AMiHRA - ℓ_1	0.65	12	80.00	**95.00**	0.885	10	**94.00**	90.00
	AMiHRA - $\ell_1-\ell_2$	0.65, 0.91, 0.25	12	**98.00**	**95.00**	0.885, 0.999, 0.0017	10	**94.00**	**95.00**
	AAMiHRA - ℓ_1	0.65	12	80.00	**95.00**	0.885	10	**94.00**	90.00
Glass Identification	AAMiHRA - $\ell_1-\ell_2$	0.65, 0.91, 0.25	12	**98.00**	**95.00**	0.885, 0.999, 0.0017	10	**94.00**	**95.00**
	TQW Alg 3.1 - ℓ_1	0.65	10	68.00	85.00	0.885	9	36.00	40.00
	DLY Alg 4-II - ℓ_1	0.65	9	0.00	0.00	0.885	9	32.00	25.00
	AKTIS Alg 1 - ℓ_1	0.65	13	16.00	5.00	0.885	13	18.00	15.00

TABLE 4. Performance results of all the algorithms on all the dataset and four activation functions. The best and suboptimal results are highlighted in bold and underlined, respectively

	wins / ties / looses		
	AMiHRA - ℓ_1 vs. TQW Alg 3.1 $-\ell_1$	AMiHRA - ℓ_1 vs. DLY Alg 4-II - ℓ_1	AMiHRA - ℓ_1 vs. AKTIS Alg 1 - ℓ_1
Training	12/0/0	12/0/0	12/0/0
Testing	12/0/0	10/1/1	12/0/0
	AAMiHRA - ℓ_1 vs. TQW Alg 3.1 $-\ell_1$	AAMiHRA - ℓ_1 vs. DLY Alg 4-II - ℓ_1	AAMiHRA - ℓ_1 vs. AKTIS Alg 1 $-\ell_1$
Training	12/0/0	12/0/0	12/0/0
Testing	12/0/0	10/1/1	12/0/0
	AMiHRA - $\ell_1-\ell_2$ vs. AMiHRA - ℓ_1		AAMiHRA - $\ell_1-\ell_2$ vs. AAMiHRA - ℓ_1
Training	6/4/2		6/5/1
Testing	5/7/0		5/7/0

TABLE 5. Number of wins, ties and looses of all the algorithms.

Remark 4.11. (1) *We adopted the Wilconxon signed-ranks and Sign test [15] as the statistical methods to compare the reported results of all the algorithms in Table 4. In accordance with the statistical analysis on these results with Wilconxon signed-ranks, it is noted from Table 5 that our proposed algorithms (i.e., AMiHRA - ℓ_1 and AAMiHRA - ℓ_1) considerably achieve better training and testing accuracies than TQW Alg $3.1-\ell_1$, DLY Alg 4-II - ℓ_1 and AKTIS Alg $1-\ell_1$. It is also found from the same table that the presence of the ℓ_2 penalty in AMiHRA - $\ell_1-\ell_2$ and AAMiHRA - $\ell_1-\ell_2$ improves their ability to achieve better and robust generalization performance than their correspondings AMiHRA - ℓ_1 and AAMiHRA - ℓ_1 in these experiments.*

(2) *On the hand, based on the null-hypothesis in the sign test [15], it is discovered that the normal distribution $h\big(\frac{h}{2}, \frac{\sqrt{h}}{2}\big)$ is obeyed by the number of wins for an algorithm and $h = (b$ datasets $\times$ d activation functions$)$. For this test, we assert that an algorithm is significantly better than the other, when its number of wins, compared to other is at least $\frac{h}{2} + Z_{m/2} \times \frac{\sqrt{h}}{2}$, where m is the assigned significant level. In all the experiments, we assigned $h = 12$ and $m = 0.1$, then $8 < \frac{12}{2} + 1.645 \times \frac{\sqrt{12}}{2} < 9$. This implies that an algorithm will be said to significantly achieves better performance, if its number of wins reaches at least 9. So, based on these facts, it is noticed from Table 5 that AMiHRA-ℓ_1 and AAMiHRA - ℓ_1 significantly achieve better performance than TQW Alg $3.1-\ell_1$, DLY Alg 4-II - ℓ_1 and AKTIS Alg $1-\ell_1$.*

(3) *Meanwhile, the number of wins of AMiHRA - $\ell_1-\ell_2$ and AAMiHRA - $\ell_1-\ell_2$ with ℓ_2 penalty when compared with their corresponding AMiHRA - ℓ_1 and AAMiHRA - ℓ_1 as shown in Table 5 are less than the least number, however, we noticed that they considerably achieve the highest number of wins in the experiments.*

5. CONCLUSION

This paper introduces two efficient Halpern-type inertial methods. The first is the alternated and multi-step inertial Halpern-type relaxed algorithm (AMiHRA) that involves three improved versions of the inertial steps, one of which is the alternated inertial step (1.8), while the others are the multi-step inertial steps (1.7), and the second is the accelerated alternated and multi-step inertial Halpern-type relaxed algorithm (AAMiHRA) that combines the three term conjugate gradient-like direction (1.10), the alternated inertial step (1.8) and the multi-step inertial step (1.7). In each of the two proposed algorithms, the monotonic self-adaptive step length criteria is used, which do not require any information about the norm of the underlying operator or the use of any line search procedure. The strong convergence theorem for each of the algorithms to a solution of problem (1.1) is formulated and proved based on the convergence theorem of the alternated inertial Halpern-type relaxed algorithm with perturbations in real

Hilbert spaces. The applications of the proposed methods in solving constrained minimization problems and classification problems based on the extreme learning machine ELM are analysed and their numerical results have been compared with the algorithms in [21, 53, 1]. In all the experiments based on ℓ_1-regularization approach, that is model (4.68), the numerical results show that the proposed algorithms (i.e., AMiHRA and AAMiHRA) are robust, computationally efficients and achieve better generalisation performance and stability than the algorithms in [21, 53, 1]. It is also noted from the results of the experiments that the proposed algorithms achieve better accuracy and stability based on the $\ell_1 - \ell_2$ hybrid regularization model, (i.e., the model (4.69)) than with ℓ_1-regularization model, (i.e., the model (4.69)).

Acknowledgments This research budget was allocated by National Science, Research and Innovation Fund (NSRF), King Mongkut's University of Technology North Bangkok (Project no. KMUTNB-FF-68-B-07). Additionally, the first author acknowledged the support of Petchra Pra Jom Klao Doctoral Research Scholarship from King Mongkut's University of Technology Thonburi with Contract No. 48/2565 and Center of Excellence in Theoretical and Computational Science (TaCS-CoE), KMUTT.

Availability of data and materials The datasets analyzed in this study are available in `https://archive.ics.uci.edu/`.

References

[1] J. Abubakar, P. Kumam, G. H. Taddele, A. H. Ibrahim and K. Sitthithakerngkiet: *Strong convergence of alternated inertial CQ relaxed method with application in signal recovery*, Comput. Appl. Math., **40** (8) (2021), Article ID: 310.

[2] A. Adamu, D. Kitkuan and T. Seangwattana: *An accelerated Halpern-type algorithm for solving variational inclusion problems with applications*, Bangmod J. Math. Comp. Sci, **8** (2022), 37–55.

[3] A. Ahmad, P. Kumam and M. H. Harbau: *Convergence theorems for common solutions of nonlinear problems and applications*, Carpathian J. Math., **40** (2), (2024) 207–241.

[4] A. Ahmad, P. Kumam, M. H. Harbau and K. Sitthithakerngkiet: *Inertial hybrid algorithm for generalized mixed equilibrium problems, zero problems, and fixed points of some nonlinear mappings in the intermediate sense*, Math. Methods Appl. Sci., **47** (11) (2024), 8527–8550.

[5] H. H. Bauschke, P. L. Combettes: *Convex analysis and monotone operator theory in Hilbert spaces*, 2nd Edition, CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC, Springer-Cham, New York (2017).

[6] A. Beck, M. Teboulle: *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., **2** (1) (2009), 183–202.

[7] C. Byrne: *Iterative oblique projection onto convex sets and the split feasibility problem*, Inverse problems, **18** (2) (2002), Article ID: 441.

[8] F. Cao, Y. Tan and M. Cai: *Sparse algorithms of random weight networks and applications*, Expert System Appl., **41** (5) (2014), 2457–2462.

[9] L.-C. Ceng, Q. H. Ansari and J.-C. Yao: *An extragradient method for solving split feasibility and fixed point problems*, Comput. Math. Appl., **64** (4) (2012), 633–642.

[10] Y. Censor, T. Elfving: *A multiprojection algorithm using bregman projections in a product space*, Numer. Algorithms, **8** (1994), 221–239.

[11] H. Che, Y. Zhuang, Y. Wang and H. Chen: *A relaxed inertial and viscosity method for split feasibility problem and applications to image recovery*, J. Global Optim., **87** (2-4) (2023), 619–639.

[12] P. Chen, J. Huang and X. Zhang: *A primal-dual fixed point algorithm for convex separable minimization with applications to image restoration*, Inverse Problems, **29** (2) (2013), Article ID: 025011.

[13] W. Cholamjiak, Y. Shehu and J. C. Yao: *Prediction of breast cancer through fast optimization techniques applied to machine learning*, Optimization, **73** (2024), 1–29.

[14] P. L. Combettes, L. E. Glaudin: *Quasi-nonexpansive iterations on the affine hull of orbits: from Mann's mean value algorithm to inertial methods*, SIAM J. Optim., **27** (4) (2017), 2356–2380.

[15] J. Demˇ sar: *Statistical comparisons of classifiers over multiple data sets*, J. Mach. Learn. Res., **7** (2006), 1–30.

[16] Q.-L. Dong, H.-b. Yuan: *Accelerated mann and cq algorithms for finding a fixed point of a nonexpansive mapping*, Fixed Point Theory Appl., **2015** (2015), Article ID: 125.

[17] Q.-L. Dong, Y. J. Cho and T. M. Rassias: *General inertial Mann algorithms and their convergence analysis for nonexpansive mappings*, in: Applications of nonlinear analysis, Springer Optim. Appl., Springer, Cham, New York, **134** (2018), 175–191.

[18] Q. L. Dong, Y. C. Tang, Y. J. Cho and T. M. Rassias,: *"Optimal" choice of the step length of the projection and contraction methods for solving the split feasibility problem*, J. Global Optim., **71** (2) (2018), 341–360.

[19] Q.-L. Dong, J. Huang, X. Li, Y.J. Cho and T. M. Rassias, *Mikm: multi-step inertial krasnosel'skiĭ–mann algorithm and its applications*, J. Global Optim., **73** (2019), 801–824.

[20] Q.-L. Dong, X.-H. Li and T. M. Rassias: *Two projection algorithms for a class of split feasibility problems with jointly constrained nash equilibrium models*, Optimization, **70** (4) (2021), 871–897.

[21] Q.-L. Dong, L. Liu and Y. Yao: *Self-adaptive projection and contraction methods with alternated inertial terms for solving the split feasibility problem*, J. Nonlinear Convex Anal., **23** (3) (2022), 591–605.

[22] B. German: *Glass Identification*, (1987), DOI: 10.24432/C5WW2P.

[23] A. Gibali, L.-W. Liu and Y.-C. Tang: *Note on the modified relaxation CQ algorithm for the split feasibility problem*, Optim. Lett., **12** (4) (2018), 817–830.

[24] A. Gibali, D. V. Thong and N. T. Vinh: Three new iterative methods for solving inclusion problems and related problems, Comput. Appl. Math., **39** (3) (2020), Article ID: 187.

[25] K. Goebel, S. Reich: *Uniform convexity, hyperbolic geometry, and nonexpansive mappings*, Monographs and Textbooks in Pure and Applied Mathematics, **83**, Marcel Dekker, Inc., New York (1984).

[26] M. H. Harbau, G. C. Ugwunnadi, L. O. Jolaoso and A. Abdulwahab: *Inertial accelerated algorithm for fixed point of asymptotically nonexpansive mapping in real uniformly convex banach spaces*, Axioms, **10** (3) (2021), Article ID: 147.

[27] M. H. Harbau, A. Ahmad, B. Ali and G. C. Ugwunnadi: *Inertial residual algorithm for fixed points of finite family of strictly pseudocontractive mappings in banach spaces*, Int. J. Nonlinear Anal. Appl., **13** (2) (2022), 2257–2269.

[28] S. He, C. Yang: Solving the variational inequality problem defined on intersection of finite level sets, in: Abstract Appl. Anal., **2013** (2013), Article ID: 942315.

[29] G.-B. Huang, Q.-Y. Zhu and C.-K. Siew: *Extreme learning machine: a new learning scheme of feedforward neural networks*, in: 2004 IEEE international joint conference on neural networks (IEEE Cat. No. 04CH37541), IEEE, **2** (2004), 985–990.

[30] H. Iiduka, I. Yamada: *A use of conjugate gradient direction for the convex optimization problem over the fixed point set of a nonexpansive mapping*, SIAM J. Optim., 19 (4) (2009) 1881–1893.

[31] H. Iiduka: *Three-term conjugate gradient method for the convex optimization problem over the fixed point set of a nonexpansive mapping*, Appl. Math. Comp., **217** (13) (2011), 6315–6327.

[32] H. Iiduka: *Fixed point optimization algorithms for distributed optimization in networked systems*, SIAM J. Optim., **23** (1) (2013), 1–26.

[33] H. Iiduka: *Acceleration method for convex optimization over the fixed point set of a nonexpansive mapping*, Math. Prog., **149** (1-2) (2015), 131–165.

[34] A. Janosi, W. Steinbrunn, M. Pfisterer and R. Detrano: *Heart Disease*, (1988), DOI: 10.24432/C52P4X.

[35] N. Jun-On, W. Cholamjiak: *Enhanced Double Inertial Forward–Backward Splitting Algorithm for Variational Inclusion Problems: Applications in Mathematical Integrated Skill Prediction*, Symmetry, **16** (8) (2024), Article ID: 1091.

[36] A. Kiri, A. Abubakar: *A family of conjugate gradient projection method for nonlinear monotone equations with applications to compressive sensing*, Nonlinear Convex Anal. Optim., **1** (1) (2022), 47–65.

[37] J. Liang: *Convergence rates of first-order operator splitting methods*, Ph.D. thesis, Normandie Université; GREYC CNRS UMR 6072 (2016).

[38] D. A. Lorenz, T. Pock: *An inertial forward-backward algorithm for monotone inclusions*, J. Math. Imaging Vision, **51** (2) (2015), 311–325.

[39] X. Ma, H. Liu: *An inertial Halpern-type CQ algorithm for solving split feasibility problems in Hilbert spaces*, J. Appl. Math. Comput., **68** (3) (2022), 1699–1717.

[40] Z. Mu, Y. Peng: *A note on the inertial proximal point method*, Stat. Optim. Inf. Comput., **3** (3) (2015), 241–248.

[41] S. Penfold, R. Zalas, M. Casiraghi, M. Brooke, Y. Censor and R. Schulte: *Sparsity constrained split feasibility for dose-volume constraints in inverse planning of intensity-modulated photon or proton therapy*, Phys. Med. Biol., **62** (9) (2017), Article ID: 3599.

[42] P. Phairatchatniyom, H. Rehman, J. Abubakar, P. Kumam and J. Mart´ınez-Moreno: *An inertial iterative scheme for solving split variational inclusion problems in real Hilbert spaces*, Bangmod J. Math. Comp. Sci., **7** (2021), 35–52.

[43] B. T. Polyak: *Some methods of speeding up the convergence of iteration methods*, USSR Comp. Math. Math. Phys., **4** (5) (1964), 1–17.

[44] B. Qu, N. Xiu: *A note on the CQ algorithm for the split feasibility problem*, Inverse Problems, **21** (5) (2005), 1655–1665.

[45] S. Reich, T. M. Tuyen and P. T. V. Huyen: *Inertial proximal point algorithms for solving a class of split feasibility problems*, J. Optim. Theory Appl., **200** (3) (2024), 951–977.

[46] D. R. Sahu, Y. J. Cho, Q.-L. Dong, M. Kashyap and X. Li: *Inertial relaxed cq algorithms for solving a split feasibility problem in hilbert spaces*, Numer. Algorithms, **87** (2021), 1075–1095.
[47] D. Serre: *Matrices*, Graduate Texts in Mathematics, **216** Springer-Verlag, New York (2002), Theory and Applications, Translated from the 2001 French original.
[48] Y. Shehu, Q.-L. Dong, L.-L. Liu: *Global and linear convergence of alternated inertial methods for split feasibility problems*, Rev. R. Acad. Cienc. Exactas Fís. Nat. Ser. A Mat., **115** (2) (2021), Article ID: 53.
[49] S. Suantai, B. Panyanak, S. Kesornprom and P. Cholamjiak: *Inertial projection and contraction methods for split feasibility problem applied to compressed sensing and image restoration*, Optim. Lett., **16** (6) (2022), 1725–1744.
[50] S. Suantai, P. Peeyada, A. Fulga and W. Cholamjiak: *Heart disease detection using inertial Mann relaxed CQ algorithms for split feasibility problems*, AIMS Math., **8** (8) (2023), 18898–18918.
[51] G. H. Taddele, A. G. Gebrie and J. Abubakar: *An iterative method with inertial effect for solving multiple-sets split feasibility problem*, Bangmod J. Math. Comp. Sci, **7** (2021), 53–73.
[52] B. Tan, X. Qin and X. Wang: *Alternated inertial algorithms for split feasibility problems*, Numer. Algorithms, **95** (2023) 773–812.
[53] B. Tan, X. Qin and X. Wang: *Alternated inertial algorithms for split feasibility problems*, Numer. Algorithms, **95** (2) (2024), 773–812.
[54] R. Tibshirani: *Regression shrinkage and selection via the lasso*, J. Roy. Statist. Soc. Ser. B, **58** (1) (1996), 267–288.
[55] W. Wolberg, O. Mangasarian, N. Street and W. Street: *Breast Cancer Wisconsin (Diagnostic)*, (1995), DOI: 10.24432/C5DW2B
[56] H.-K. Xu: *Iterative methods for the split feasibility problem in infinite-dimensional Hilbert spaces*, Inverse Problems, **26** (10) (2010), Artile ID: 105018.
[57] W. Yajai, P. Nabheerong and W. Cholamjiak: *A double inertial Mann Algorithm for Split Equilibrium Problems Application to Breast Cancer Screening*, J. Nonlinear Convex Anal., **25** (7) (2024), 1697–1716.
[58] W. Yajai, S. Das, S. Yajai and W. Cholamjiak: *A modified viscosity type inertial subgradient extragradient algorithm for nonmonotone equilibrium problems and application to cardiovascular disease detection*, Discrete Contin. Dyn. Syst. Ser. S, (2024), DOI: 10.3934/dcdss.2024163
[59] Q. Yang: *The relaxed cq algorithm solving the split feasibility problem*, Inverse problems, **20** (4) (2004), Article ID: 1261.
[60] Q. Yang: *On variable-step relaxed projection algorithm for variational inequalities*, J. Math. Anal. Appl., **302** (1) (2005), 166–179.
[61] H. Ye, F. Cao and D. Wang: *A hybrid regularization approach for random vector functional-link networks*, Expert Systems Appl., **140** (2020) Article ID: 112912.

ABDULWAHAB AHMAD
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY THONBURI (KMUTT)
CENTER OF EXCELLENCE IN THEORETICAL AND COMPUTATIONAL SCIENCE (TACS-COE) & FIXED POINT RESEARCH LABORATORY, ROOM SCL 802, SCIENCE LABORATORY BUILDING, DEPARTMENTS OF MATHEMATICS, FACULTY OF SCIENCE
126 PRACHA-UTHIT ROAD THUNG KHRU, BANG MOD 10140, BANGKOK, THAILAND
ORCID: 0009-0000-0460-0204
Email address: abumuhammaddm@gmail.com

POOM KUMAM
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY THONBURI (KMUTT)
CENTER OF EXCELLENCE IN THEORETICAL AND COMPUTATIONAL SCIENCE (TACS-COE) & FIXED POINT RESEARCH LABORATORY, ROOM SCL 802, SCIENCE LABORATORY BUILDING, DEPARTMENTS OF MATHEMATICS, FACULTY OF SCIENCE
126 PRACHA-UTHIT ROAD THUNG KHRU, BANG MOD 10140, BANGKOK, THAILAND
ORCID: 0000-0002-5463-4581
Email address: poom.kum@kmutt.ac.th

YEOL JE CHO
GYEONGSANG NATIONAL UNIVERSITY
DEPARTMENT OF MATHEMATICS EDUCATION
52828, JINJU, KOREA
ORCID: 0000-0002-1250-2214
Email address: yjchomat@gmail.com

Kanokwan Sitthithakerngkiet
Intelligent and Nonlinear Dynamic Innovations Research Center
Department of Mathematics, Faculty of Applied Sciences
King Mongkut's University of Technology, North Bangkok, (KMUTNB)
Wongsawang, Bangsue, 1518, Bangkok 10800, Thailand
ORCID: 0000-0002-8496-7803
Email address: kanokwan.s@sci.kmutnb.ac.th