CARPATHIAN J. MATH.
Volume **40** (2024), No. 3,
Pages 737 - 752

Online version at https://www.carpathian.cunbm.utcluj.ro/
Print Edition: ISSN 1584 - 2851; Online Edition: ISSN 1843 - 4401
DOI: https://doi.org/10.37193/CJM.2024.03.13

A fast contraction algorithm using two inertial extrapolations for variational inclusion problem and data classification

SUTHEP SUANTAI[1], PRASIT CHOLAMJIAK[3], PAPATSARA INKRONG[3] and SUPARAT KESORNPROM[2*]

ABSTRACT. In this paper, we propose a new method for solving variational inclusion problems in Hilbert spaces. This algorithm uses two inertial terms to speed up the convergence. In order to avoid computing the Lipschitz stepsize, we use an updated stepsize which is not necessary to know the Lipschitz constant of the operator. The weak convergence is established under some mild conditions. We present numerical performance of the proposed algorithm and compare our algorithm with other algorithms in literature. Finally, we deduce our algorithm for solving the convex minimization problem and give an application to the data classification problem of heart failure dataset.

1. INTRODUCTION

Let H be a real Hilbert space, $A : H \to 2^H$ be a set-valued mapping and $f : H \to H$ be a single-valued nonlinear mapping. We consider the following variational inclusion problem (VIP):

$$\text{find a point } x^* \in H \text{ such that } 0 \in A(x^*) + f(x^*). \tag{1.1}$$

If $f \equiv 0$, then VIP reduces to the inclusion problem [25] which is finding a point $x^* \in H$ such that

$$0 \in A(x^*). \tag{1.2}$$

We observe that VIP is general in the sense that it includes optimization problem, optimal control, mathematical programming and so on, see [1, 6, 13, 20, 24, 30]. Moudafi [20] showed that x^* is a solution of (1.1) if and only if $x^* = J_\lambda^A(I - \lambda f)(x^*)$, for all $\lambda > 0$, where $J_\lambda^A : H \to H$ is the resolvent operator associated with A and λ defined by

$$J_\lambda^A(x) = (I + \lambda A)^{-1}(x),\ x \in H. \tag{1.3}$$

We know that J_λ^A is a single-valued and nonexpansive mapping.

In recent years, several authors studied and paid their attentions on VIP and provided various iterative algorithms for solving such problem, see for examples, [8, 12, 17, 27, 28, 34] and the reference therein. A popular one was introduced by Rockafellar [25] which is called the proximal point algorithm (PPA):

$$x_{k+1} = J_{\lambda_k}^A(x_k),\ \lambda_k \subset (0, +\infty). \tag{1.4}$$

Received: 24.11.2023. In revised form: 15.03.2024. Accepted: 22.03.2024
2010 *Mathematics Subject Classification.* 49J40, 47J20, 35A15, 65K10, 97C70, 97U70.
Key words and phrases. *variational inclusion problem, monotone mapping, inertial method, weak convergence, data classification.*
Corresponding author: Suparat Kesornprom; suparat.ke@gmail.com

In 2001, Alvarez and Attouch [2] introduced the inertial proximal point algorithm (iPPA) for a maximal monotone operator as follows: $x_0, x_1 \in H$ and

$$x_{k+1} = J^A_{\lambda_k}(x_k + \alpha_k(x_k - x_{k-1})), \tag{1.5}$$

where $\alpha_k \in [0,1]$. They proved weak convergence theorem by assuming $\sum_{k=1}^{\infty} \alpha_k \|x_k - x_{k-1}\|^2 < +\infty$. The extrapolation term $\alpha_k(x_k - x_{k-1})$ can accelerate the convergence properties; see [4, 9, 23].

In 2018, Dong et al. [11] introduced a general inertial Mann algorithm to accelerate Mann algorithm as follows: $x_0, x_1 \in H$ and

$$\begin{aligned} w_k &= x_k + \alpha_k(x_k - x_{k-1}), \\ y_k &= x_k + \delta_k(x_k - x_{k-1}), \\ x_{k+1} &= (1-\lambda_k)w_k + \lambda_k T(y_k), \end{aligned} \tag{1.6}$$

where $T : H \to H$ is a nonexpansive mapping and the sequences $\{\alpha_k\}$, $\{\delta_k\}$ and $\{\lambda_k\}$ satisfy conditions in [11]. Recently, several authors studied two-step inertial methods and multi-step inertial methods (see for example [15, 18, 19, 26, 32, 33]).

In 2014, Cai et al. [7] studied convergence rate of the projection and contraction algorithms for variational inequalities.

In 2018, Zhang and Wang [31] introduced a contraction algorithm by combining optimal step as follows:

$$\begin{aligned} y_k &= J^A_{\lambda_k}(x_k - \lambda_k f(x_k)) \\ x_{k+1} &= x_k - \gamma\beta_k d(x_k, y_k), \end{aligned}$$

where

$$\begin{aligned} d(x_k, y_k) &= (x_k - y_k) - \lambda_k(f(x_k) - f(y_k)), \\ \beta_k &= \frac{\phi(x_k, y_k)}{\|d(x_k, y_k)\|^2}, \end{aligned}$$

and

$$\phi(x_k, y_k) = \langle x_k - y_k, d(x_k, y_k)\rangle, \tag{1.7}$$

where $\gamma \in (0,2)$ and the stepsize λ_k satisfies the conditions in [31]. They proved weak convergence theorems for solving VIP.

In 2023, Dey [10] introduced a hybrid inertial and contraction proximal point algorithm for a monotone variational inclusion as follows: $x_0, x_1 \in H$ and

$$\begin{aligned} \bar{\alpha_k} &= \begin{cases} \min\{\alpha, \frac{\tau_k}{\|x_k - x_{k-1}\|}\} & \text{if } x_k \neq x_{k-1} \\ \alpha & \text{otherwise} \end{cases} \\ w_k &= x_k + \alpha_k(x_k - x_{k-1}), \\ y_k &= J^A_{\lambda_k}(w_k - \lambda_k f(w_k)) \\ z_k &= w_k - \gamma\beta_k d(w_k, y_k) \\ d(w_k, y_k) &= (w_k - y_k) - \lambda_k(f(w_k) - f(y_k)) \\ \phi(w_k, y_k) &= \langle w_k - y_k, d(w_k, y_k)\rangle \\ \beta_k &= \begin{cases} \frac{\phi(w_k, y_k)}{\|d(w_k, y_k)\|^2} & \text{if } d(w_k, y_k) \neq 0 \\ 0 & \text{if } d(w_k, y_k) = 0 \end{cases} \\ x_{k+1} &= (1 - \theta_k - \eta_k)x_k + \theta_k z_k \end{aligned} \tag{1.8}$$

where $\alpha > 0$, $\gamma \in (0,2)$ and $\{\lambda_k\}$, $\{\tau_k\}$, $\{\theta_k\}$ and $\{\eta_k\}$ are defined as in [10].

According to the algorithms of Zhang and Wang [31] and Dey [10], it depends on the Lipschitz constant, which is generally not easy to compute in practice. In 2021, Hieu et al. [30] used the stepsize which is updated over each iteration. These stepsizes are not necessary to know the Lipschitz constant of the operator.

In this paper, we design and modify a contraction algorithm by combining the optimal step with inertial terms and updated stepsize which are introduced by Dong et al. [11] and Hieu et al. [30], respectively. We prove a weak convergence theorem for solving the variational inclusion problem in Hilbert spaces. Numerical examples in finite dimensional spaces are presented to show the efficiency of our algorithm and to compare it with algorithms in literature review. Moreover, the proposed algorithm is applied to solve the convex minimization problem and the data classification problem.

This paper is organized as follows. In Section 2, we provide some basic preliminaries. In Section 3, we introduce a new algorithm and prove the weak convergence theorem in Hilbert spaces. In Section 4, we present numerical examples in finite dimensional spaces. In Section 5, we provide applications to solve the convex minimization problem and the data classification problem. We finally give conclusion in Section 6.

2. Preliminaries and lemmas

In this section, we provide some basic definitions and lemmas which will be used in the sequel. Let H be a real Hilbert space. In what follows, we use the following notations:

- the symbol $\rightharpoonup$ stands for the weak convergence.
- the symbol $\to$ stands for the strong convergence.

Recall that a mapping $T : H \to H$ is said to be

(1) nonexpansive if
$$\|Tx - Ty\| \leq \|x - y\|, \ \forall x, y \in H.$$

(2) firmly-nonexpansive if
$$\langle Tx - Ty, x - y\rangle \geq \|Tx - Ty\|^2, \ \forall x, y \in H.$$

We note that if T is firmly-nonexpansive, then $I - T$ is also firmly-nonexpansive.

(3) L-Lipschitz continuous, if there exists a constant $L > 0$ such that
$$\|Tx - Ty\| \leq L\|x - y\|, \ \forall x, y \in H.$$

(4) monotone if for all $x, y \in H$,
$$\langle Tx - Ty, x - y\rangle \geq 0.$$

(5) A multi-valued mapping $B : H \to 2^H$ is called monotone if for all $x, y \in H$
$$\langle u - v, x - y\rangle \geq 0, \text{ for all } (x, u), (y, v) \in G(A),$$
where its graph is defined by
$$G(A) = \{(x, y) \in H \times H : y \in A(x)\}.$$

(6) A multi-valued mapping $A : H \to 2^H$ is maximally monotone if its graph is not properly contained in the graph of any other monotone operators.

It is well-known that A is maximally monotone if and only if for $(x, y) \in H \times H$, $\langle x - v, y - w\rangle \geq 0$ for every $(v, w) \in G(A)$ implies $y \in A(x)$.

Lemma 2.1. [3] *Let $A : H \to 2^H$ be a maximal monotone mapping and let $f : H \to H$ be a Lipschitz continuous mapping. Then the mapping $A + f$ is a maximal monotone mapping.*

Lemma 2.2. *(Demiclosedness principle*[14]*) Let C be a nonempty closed convex subset of a real Hilbert space H and let $T : C \to C$ be a nonexpansive mapping. If $x_k \rightharpoonup x \in C$ and $\lim_{k\to\infty} \|x_k - Tx_k\| = 0$, then $x = Tx$.*

Lemma 2.3. [22] *Let* $\{a_k\}$, $\{b_k\}$ *and* $\{c_k\}$ *be real positive sequences such that*

$$a_{k+1} \leq (1 + c_k)a_k + b_k,\ k \geq 1.$$

If $\sum_{k=1}^{\infty} c_k < +\infty$ *and* $\sum_{k=1}^{\infty} b_k < +\infty$, *then* $\lim\limits_{k\to+\infty} a_k$ *exists.*

Lemma 2.4. *(Opial theorem* [21]*) Let* C *be a nonempty subset of a real Hilbert space* H *and* $\{x_k\}$ *be a sequence in* H *that satisfies the following properties:*

(i) $\lim\limits_{k\to\infty} \|x_k - x\|$ *exists for each* $x \in C$;

(ii) every weak sequential cluster point of $\{x_k\}$ *belongs to* C.

Then $\{x_k\}$ *converges weakly to a point in* C.

3. Main results

This section presents a fast contraction algorithm for solving VIP. We next introduce the following lemma for proving our theorem.

Lemma 3.5. *Let* $\varphi_{-1} \geq 0, \varphi_0 \geq 0$ *and* $\{\varphi_k\}$, $\{\eta_k\}$ *and* $\{\delta_k\}$ *be nonnegative real sequences satisfying*

$$\varphi_{k+1} \leq (1 + \eta_k)\varphi_k + (\eta_k + \delta_k)\varphi_{k-1} + \delta_k\varphi_{k-2},\ k \geq 1. \tag{3.9}$$

Then

$$\varphi_{k+1} \leq M \cdot \prod_{j=1}^{k}(1 + 2\eta_j + 2\delta_j), \tag{3.10}$$

where $M = \max\{\varphi_{-1}, \varphi_0, \varphi_1\}$. *Furthermore, if* $\sum_{k=1}^{\infty} \eta_k < +\infty$ *and* $\sum_{k=1}^{\infty} \delta_k < +\infty$, *then* $\{\varphi_k\}$ *is bounded.*

Proof. By using mathematical induction, we can prove this lemma. See also [16]. □

In this work, we assume the following conditions to obtain the weak convergence of our algorithm.

Condition (i) The solution set Φ of VIP (1.1) is nonempty.

Condition (ii) The mapping f is monotone and Lipschitz continuous.

Condition (iii) The mapping A is maximally monotone.

Algorithm 3.1. *Suppose that* $\{\eta_k\}$ *and* $\{\delta_k\}$ *are nonnegative sequences satisfying* $\sum_{k=1}^{\infty} \eta_k < +\infty$ *and* $\sum_{k=1}^{\infty} \delta_k < +\infty$. *Let* $\gamma \in (0, 2)$, $\lambda_0 > 0$, $\mu \in (0, 1)$ *and* x_{-1}, x_0 *and* x_1 *be chosen arbitrary. Calculate* x_{k+1} *as follows:*

$$w_k = x_k + \eta_k(x_k - x_{k-1}) + \delta_k(x_{k-1} - x_{k-2}) \tag{3.11}$$

$$y_k = J^A_{\lambda_k}(w_k - \lambda_k f(w_k)) \tag{3.12}$$

$$d(w_k, y_k) = (w_k - y_k) - \lambda_k(f(w_k) - f(y_k)) \tag{3.13}$$

$$x_{k+1} = w_k - \gamma\beta_k d(w_k, y_k), \tag{3.14}$$

where

$$\lambda_{k+1} = \begin{cases} \min\{\frac{\mu\|w_k - y_k\|}{\|f(w_k) - f(y_k)\|}, \lambda_k\} & \text{if } f(w_k) - f(y_k) \neq 0 \\ \lambda_k & \text{otherwise} \end{cases} \tag{3.15}$$

and

$$\beta_k = \frac{\phi(w_k, y_k)}{\|d(w_k, y_k)\|^2},\ \phi(w_k, y_k) = \langle w_k - y_k, d(w_k, y_k)\rangle. \tag{3.16}$$

Remark 3.1. It is easy to see that the sequence $\{\lambda_k\}$ is non-increasing. Since f is Lipschitz continuous, there exists $L > 0$ such that $\|f(w_k) - f(y_k)\| \le L\|w_k - y_k\|$. Hence,

$$\lambda_{k+1} = \min\left\{\frac{\mu\|w_k - y_k\|}{\|f(w_k) - f(y_k)\|}, \lambda_k\right\} \ge \min\{\frac{\mu}{L}, \lambda_k\}. \tag{3.17}$$

By the definition of $\{\lambda_k\}$, it implies that the sequence $\{\lambda_k\}$ is bounded from below by $\min\{\lambda_0, \frac{\mu}{L}\}$. So, we obtain $\lim_{k\to\infty} \lambda_k = \lambda > 0$.

Lemma 3.6. *In (3.12), if $w_k = y_k$ for some k, then $w_k \in \Phi$.*

Proof. If $w_k = y_k$, then $w_k = J^A_{\lambda_k}(w_k - \lambda_k f(w_k))$. It follows that

$$\begin{aligned} w_k = (I + \lambda_k A)^{-1}(w_k - \lambda_k f(w_k)) \quad &\Leftrightarrow \quad w_k - \lambda_k f(w_k) \in w_k + \lambda_k A w_k \\ &\Leftrightarrow \quad -f(w_k) \in A w_k \\ &\Leftrightarrow \quad 0 \in A w_k + f(w_k). \end{aligned} \tag{3.18}$$

Hence $w_k \in \Phi$. □

Lemma 3.7. *Let $x^* \in \Phi$. Assume that the sequence $\{x_k\}$ generated by Algorithm 3.1. If $\{\lambda_k\}$ satisfies (3.15), then under Conditions (i), (ii) and (iii), we have*

$$\begin{aligned} \|x_{k+1} - x^*\|^2 \quad \le \quad & \|x_k - x^*\|^2 + \eta_k^2\|x_k - x_{k-1}\|^2 + 2\eta_k\|x_k - x^*\|\|x_k - x_{k-1}\| \\ & + \delta_k^2\|x_{k-1} - x_{k-2}\|^2 + 2\delta_k\|x_k - x^*\|\|x_{k-1} - x_{k-2}\| \\ & + 2\eta_k\delta_k\|x_k - x_{k-1}\|\|x_{k-1} - x_{k-2}\| \\ & - \gamma(2-\gamma)\beta_k^2\|d(w_k, y_k)\|^2. \end{aligned} \tag{3.19}$$

Proof. By definition of x_{k+1}, we have

$$\begin{aligned} \|x_{k+1} - x^*\|^2 \quad &= \quad \|w_k - \gamma\beta_k d(w_k, y_k) - x^*\|^2 \\ &= \quad \|w_k - x^*\|^2 - 2\gamma\beta_k\langle w_k - x^*, d(w_k, y_k)\rangle + \gamma^2\beta_k^2\|d(w_k, y_k)\|^2. \end{aligned} \tag{3.20}$$

Since $J^A_{\lambda_k}$ is firmly-nonexpansive and $J^A_{\lambda_k}(I - \lambda_k f)x^* = x^*$, it follows that

$$\begin{aligned} & \langle J^A_{\lambda_k}(I - \lambda_k f)w_k - J^A_{\lambda_k}(I - \lambda_k f)x^*, (I - \lambda_k f)w_k - (I - \lambda_k f)x^*\rangle \\ \ge \quad & \|J^A_{\lambda_k}(I - \lambda_k f)w_k - J^A_{\lambda_k}(I - \lambda_k f)x^*\|^2 \\ = \quad & \|y_k - x^*\|^2. \end{aligned} \tag{3.21}$$

From (3.21), we have

$$\begin{aligned} & \langle y_k - x^*, w_k - y_k - \lambda_k f(w_k)\rangle \\ = \quad & \langle J^A_{\lambda_k}(I - \lambda_k f)w_k - J^A_{\lambda_k}(I - \lambda_k f)x^*, w_k - y_k - x^* + x^* \\ & -\lambda_k f(x^*) + \lambda_k f(x^*) - \lambda_k f(w_k)\rangle \\ = \quad & \langle J^A_{\lambda_k}(I - \lambda_k f)w_k - J^A_{\lambda_k}(I - \lambda_k f)x^*, w_k - \lambda_k f(w_k) - x^* \\ & +\lambda_k f(x^*) + x^* - \lambda_k f(x^*) - y_k\rangle \\ = \quad & \langle J^A_{\lambda_k}(I - \lambda_k f)w_k - J^A_{\lambda_k}(I - \lambda_k f)x^*, (I - \lambda_k f)w_k - (I - \lambda_k f)x^* \\ & +(I - \lambda_k f)x^* - y_k\rangle \\ = \quad & \langle J^A_{\lambda_k}(I - \lambda_k f)w_k - J^A_{\lambda_k}(I - \lambda_k f)x^*, (I - \lambda_k f)w_k - (I - \lambda_k f)x^*\rangle \\ & +\langle J^A_{\lambda_k}(I - \lambda_k f)w_k - J^A_{\lambda_k}(I - \lambda_k f)x^*, (I - \lambda_k f)x^* - y_k\rangle \\ \ge \quad & \|y_k - x^*\|^2 + \langle y_k - x^*, x^* - \lambda_k f(x^*) - y_k\rangle \\ = \quad & -\langle y_k - x^*, \lambda_k f(x^*)\rangle. \end{aligned} \tag{3.22}$$

From (3.22), we obtain

$$\begin{aligned} & \langle y_k - x^*, w_k - y_k - \lambda_k(f(w_k) - f(x^*))\rangle \\ = & \langle y_k - x^*, w_k - y_k - \lambda_k f(w_k)\rangle + \langle y_k - x^*, \lambda_k f(x^*)\rangle \\ \geq & -\langle y_k - x^*, \lambda_k f(x^*)\rangle + \langle y_k - x^*, \lambda_k f(x^*)\rangle \\ = & 0. \end{aligned} \tag{3.23}$$

By the monotonicity of f and $\lambda_k > 0$, we see that

$$\langle y_k - x^*, \lambda_k f(y_k) - \lambda_k f(x^*)\rangle \geq 0. \tag{3.24}$$

Combining (3.23) and (3.24), we have

$$\begin{aligned} & \langle y_k - x^*, w_k - y_k - \lambda_k(f(w_k) - f(y_k))\rangle \\ = & \langle y_k - x^*, d(w_k, y_k)\rangle \\ \geq & 0. \end{aligned} \tag{3.25}$$

So, from (3.25), we obtain

$$\begin{aligned} \langle w_k - x^*, d(w_k, y_k)\rangle &= \langle w_k - y_k, d(w_k, y_k)\rangle + \langle y_k - x^*, d(w_k, y_k)\rangle \\ &\geq \langle w_k - y_k, d(w_k, y_k)\rangle \\ &= \phi(w_k, y_k). \end{aligned} \tag{3.26}$$

From (3.20), (3.26) and definition of β_k, we have

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &= \|w_k - x^*\|^2 - 2\gamma\beta_k\langle w_k - x^*, d(w_k, y_k)\rangle + \gamma^2\beta_k^2\|d(w_k, y_k)\|^2 \\ &\leq \|w_k - x^*\|^2 - 2\gamma\beta_k\phi(w_k, y_k) + \gamma^2\beta_k^2\|d(w_k, y_k)\|^2 \\ &= \|w_k - x^*\|^2 - 2\gamma\beta_k\frac{\phi(w_k, y_k)}{\|d(w_k, y_k)\|^2}\|d(w_k, y_k)\|^2 + \gamma^2\beta_k^2\|d(w_k, y_k)\|^2 \\ &= \|w_k - x^*\|^2 - 2\gamma\beta_k^2\|d(w_k, y_k)\|^2 + \gamma^2\beta_k^2\|d(w_k, y_k)\|^2 \\ &= \|w_k - x^*\|^2 - \gamma(2-\gamma)\beta_k^2\|d(w_k, y_k)\|^2. \end{aligned} \tag{3.27}$$

Consider,

$$\begin{aligned} \|w_k - x^*\|^2 &= \|x_k + \eta_k(x_k - x_{k-1}) + \delta_k(x_{k-1} - x_{k-2}) - x^*\|^2 \\ &= \|x_k - x^* + \eta_k(x_k - x_{k-1})\|^2 + \delta_k^2\|x_{k-1} - x_{k-2}\|^2 \\ &\quad +2\langle x_k - x^* + \eta_k(x_k - x_{k-1}), \delta_k(x_{k-1} - x_{k-2})\rangle \\ &= \|x_k - x^*\|^2 + \eta_k^2\|x_k - x_{k-1}\|^2 + 2\langle x_k - x^*, \eta_k(x_k - x_{k-1})\rangle \\ &\quad +\delta_k^2\|x_{k-1} - x_{k-2}\|^2 + 2\langle x_k - x^*, \delta_k(x_{k-1} - x_{k-2})\rangle \\ &\quad +2\langle \eta_k(x_k - x_{k-1}), \delta_k(x_{k-1} - x_{k-2})\rangle \\ &\leq \|x_k - x^*\|^2 + \eta_k^2\|x_k - x_{k-1}\|^2 + 2\eta_k\|x_k - x^*\|\|x_k - x_{k-1}\| \\ &\quad +\delta_k^2\|x_{k-1} - x_{k-2}\|^2 + 2\delta_k\|x_k - x^*\|\|x_{k-1} - x_{k-2}\| \\ &\quad +2\eta_k\delta_k\|x_k - x_{k-1}\|\|x_{k-1} - x_{k-2}\|. \end{aligned} \tag{3.28}$$

From (3.27) and (3.28), we obtain

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &\leq \|x_k - x^*\|^2 + \eta_k^2\|x_k - x_{k-1}\|^2 + 2\eta_k\|x_k - x^*\|\|x_k - x_{k-1}\| \\ &\quad +\delta_k^2\|x_{k-1} - x_{k-2}\|^2 + 2\delta_k\|x_k - x^*\|\|x_{k-1} - x_{k-2}\| \\ &\quad +2\eta_k\delta_k\|x_k - x_{k-1}\|\|x_{k-1} - x_{k-2}\| \\ &\quad -\gamma(2-\gamma)\beta_k^2\|d(w_k, y_k)\|^2. \end{aligned} \tag{3.29}$$

□

Lemma 3.8. *Let $x^* \in \Phi$. Assume that $\{x_k\}$ is generated by Algorithm 3.1. Then $\lim_{k\to\infty} \|x_k - x^*\|$ exists.*

Proof. By definition of w_k, we see that

$$\begin{aligned} \|w_k - x^*\| &= \|x_k + \eta_k(x_k - x_{k-1}) + \delta_k(x_{k-1} - x_{k-2}) - x^*\| \\ &\leq \|x_k - x^*\| + \eta_k\|x_k - x_{k-1}\| + \delta_k\|x_{k-1} - x_{k-2}\|. \end{aligned} \tag{3.30}$$

From (3.27) and (3.30), it follows that

$$\begin{aligned} \|x_{k+1} - x^*\| &\leq \|x_k - x^*\| + \eta_k\|x_k - x_{k-1}\| + \delta_k\|x_{k-1} - x_{k-2}\| \\ &\leq \|x_k - x^*\| + \eta_k(\|x_k - x^*\| + \|x_{k-1} - x^*\|) \\ &\quad +\delta_k(\|x_{k-1} - x^*\| + \|x_{k-2} - x^*\|) \\ &= (1+\eta_k)\|x_k - x^*\| + (\eta_k + \delta_k)\|x_{k-1} - x^*\| + \delta_k\|x_{k-2} - x^*\|. \end{aligned} \tag{3.31}$$

Using Lemma 3.5, we conclude that

$$\|x_{k+1} - x^*\| \leq M \prod_{j=1}^{k}(1 + 2\eta_j + 2\delta_j), \tag{3.32}$$

where $M = \max\{\|x_1 - x^*\|, \|x_0 - x^*\|, \|x_{-1} - x^*\|\}$. Moreover, by Lemma 3.5, we also have $\{x_k\}$ is bounded. Hence, $\sum_{k=1}^{\infty} \eta_k\|x_k - x_{k-1}\| < +\infty$ and $\sum_{k=1}^{\infty} \delta_k\|x_{k-1} - x_{k-2}\| < +\infty$. Using Lemma 2.3 and (3.31), it shows that $\lim_{k\to\infty} \|x_k - x^*\|$ exists. □

Lemma 3.9. *Let $x^* \in \Phi$. Assume that $\{x_k\}$ is generated by Algorithm 3.1. Then $\lim_{k\to\infty} \|w_k - y_k\| = 0$.*

Proof. From (3.15), we can see that

$$\lambda_{k+1} = \min\{\frac{\mu\|w_k - y_k\|}{\|f(w_k) - f(y_k)\|}, \lambda_k\} \leq \frac{\mu\|w_k - y_k\|}{\|f(w_k) - f(y_k)\|}. \tag{3.33}$$

It follows that

$$\|f(w_k) - f(y_k)\| \leq \frac{\mu}{\lambda_{k+1}}\|w_k - y_k\|. \tag{3.34}$$

From definition of $d(w_k, y_k)$ and (3.34), we obtain

$$\begin{aligned} \|d(w_k, y_k)\|^2 &= \|(w_k - y_k) - \lambda_k(f(w_k) - f(y_k))\|^2 \\ &= \|w_k - y_k\|^2 + \lambda_k^2\|f(w_k) - f(y_k)\|^2 - 2\lambda_k\langle w_k - y_k, f(w_k) - f(y_k)\rangle \\ &\leq \|w_k - y_k\|^2 + \frac{\mu^2\lambda_k^2}{\lambda_{k+1}^2}\|w_k - y_k\|^2 + 2\lambda_k|\langle w_k - y_k, f(w_k) - f(y_k)\rangle| \\ &\leq \|w_k - y_k\|^2 + \frac{\mu^2\lambda_k^2}{\lambda_{k+1}^2}\|w_k - y_k\|^2 + 2\lambda_k\|w_k - y_k\|\|f(w_k) - f(y_k)\| \\ &\leq \|w_k - y_k\|^2 + \frac{\mu^2\lambda_k^2}{\lambda_{k+1}^2}\|w_k - y_k\|^2 + \frac{2\mu\lambda_k}{\lambda_{k+1}}\|w_k - y_k\|^2 \\ &= (1 + \frac{\mu^2\lambda_k^2}{\lambda_{k+1}^2} + \frac{2\mu\lambda_k}{\lambda_{k+1}})\|w_k - y_k\|^2 \\ &= (1 + \frac{\mu\lambda_k}{\lambda_{k+1}})^2\|w_k - y_k\|^2. \end{aligned} \tag{3.35}$$

From definition of $\phi(w_k, y_k)$ and (3.34), we have

$$\begin{aligned}
\phi(w_k, y_k) &= \langle w_k - y_k, d(w_k, y_k)\rangle \\
&= \langle w_k - y_k, (w_k - y_k) - \lambda_k(f(w_k) - f(y_k))\rangle \\
&= \|w_k - y_k\|^2 - \lambda_k\langle w_k - y_k, f(w_k) - f(y_k)\rangle \\
&\geq \|w_k - y_k\|^2 - \lambda_k\|w_k - y_k\|\|f(w_k) - f(y_k)\| \\
&\geq \|w_k - y_k\|^2 - \frac{\mu\lambda_k}{\lambda_{k+1}}\|w_k - y_k\|^2 \\
&= (1 - \frac{\mu\lambda_k}{\lambda_{k+1}})\|w_k - y_k\|^2. \qquad (3.36)
\end{aligned}$$

Combining (3.35) and (3.36), we have

$$\begin{aligned}
\beta_k = \frac{\phi(w_k, y_k)}{\|d(w_k, y_k)\|^2} &= \frac{\langle w_k - y_k, d(w_k, y_k)\rangle}{\|d(w_k, y_k)\|^2} \\
&\geq \frac{(1 - \frac{\mu\lambda_k}{\lambda_{k+1}})}{(1 + \frac{\mu\lambda_k}{\lambda_{k+1}})^2}. \qquad (3.37)
\end{aligned}$$

Hence, using Lemma 3.7 and Lemma 3.8, we can show that

$$\lim_{k\to\infty} \|d(w_k, y_k)\| = 0. \qquad (3.38)$$

By (3.36), we have

$$\lim_{k\to\infty} \|w_k - y_k\| = 0. \qquad (3.39)$$

□

Theorem 3.1. *Let $x^* \in \Phi$. Assume that Conditions (i), (ii) and (iii) hold. Then the sequence $\{x_k\}$ generated by Algorithm 3.1 weakly converges to a point in Φ.*

Proof. Let $x^* \in \Phi$. From (3.11), we have

$$\lim_{k\to\infty} \|x_k - w_k\| = 0. \qquad (3.40)$$

Using Lemma 3.9 and (3.40), we obtain

$$\lim_{k\to\infty} \|x_k - y_k\| = 0. \qquad (3.41)$$

Let $\hat{x}$ be a weak cluster point of $\{x_k\}$. Then there exists a subsequence $\{x_{k_i}\}$ of $\{x_k\}$ such that $x_{k_i} \rightharpoonup \hat{x}$ as $i \to \infty$. Also from (3.41), we get $y_{k_i} \rightharpoonup \hat{x}$ as $i \to \infty$.

Next, we will show that $\hat{x}$ is in Φ. We know that f is Lipschitz continuous. From Lemma 2.1, we know that $A + f$ is maximally monotone. Let $(v, u) \in G(A + f)$, that is, $u - f(v) \in A(v)$. Since $y_{k_i} = J^A_{\lambda_{k_i}}(w_{k_i} - \lambda_{k_i} f(w_{k_i}))$, we get

$$w_{k_i} - \lambda_{k_i} f(w_{k_i}) \in (I + \lambda_{k_i} A)(y_{k_i}) \qquad (3.42)$$

that is,

$$\frac{w_{k_i} - y_{k_i} - \lambda_{k_i} f(w_{k_i})}{\lambda_{k_i}} \in Ay_{k_i}. \qquad (3.43)$$

Since A is maximally monotone, we have

$$\langle v - y_{k_i}, u - f(v) - \frac{w_{k_i} - y_{k_i} - \lambda_{k_i} f(w_{k_i})}{\lambda_{k_i}}\rangle \geq 0. \qquad (3.44)$$

Hence,

$$\begin{aligned} \langle v - y_{k_i}, u \rangle &\geq \langle v - y_{k_i}, f(v) + \frac{w_{k_i} - y_{k_i} - \lambda_{k_i} f(w_{k_i})}{\lambda_{k_i}} \rangle \\ &= \langle v - y_{k_i}, f(v) - f(y_{k_i}) + f(y_{k_i}) - f(w_{k_i}) + \frac{w_{k_i} - y_{k_i}}{\lambda_{k_i}} \rangle \\ &\geq \langle v - y_{k_i}, f(y_{k_i}) - f(w_{k_i}) \rangle + \langle v - y_{k_i}, \frac{w_{k_i} - y_{k_i}}{\lambda_{k_i}} \rangle. \end{aligned} \tag{3.45}$$

Since f is Lipschitz continuous and $\lim_{i\to\infty} \|w_{k_i} - y_{k_i}\| = 0$, we obtain

$$\lim_{i\to\infty} \|f(y_{k_i}) - f(w_{k_i})\| = 0. \tag{3.46}$$

Since $\lim_{i\to\infty} \lambda_{k_i} = \lambda > 0$, it follows from (3.45) that

$$\lim_{i\to\infty} \langle v - y_{k_i}, u \rangle = \langle v - \hat{x}, u \rangle \geq 0. \tag{3.47}$$

Since $A + f$ is maximally monotone, we get $0 \in (A + f)(\hat{x})$. Hence $\hat{x} \in \Phi$. Using Lemma 2.4, we conclude that $\{x_k\}$ converges weakly to a point in Φ. We complete the proof. □

4. Numerical examples

In this section, we present numerical examples to show the efficiency of Algorithm 3.1 and compare with Algorithm 1 in [10] and Algorithm 3.1 in [31].

Let $H = \mathbb{R}^n$, $f = Z^T Z$, where $Z = (z_{ij})_{n\times n}$ with randomly generated $z_{ij} \in [1, 100]$. It is well-known that f is monotone and Lipschitz continuous with Lipschitz constant $L = \|f\|^2$. We take the initial points $x_{-1} = (c_i) \in \mathbb{R}^n$, $x_0 = (d_i) \in \mathbb{R}^n$ and $x_1 = (e_i) \in \mathbb{R}^n$ where $c_i, d_i, e_i \in [0, 1]$ are generated randomly. Let A be an upper triangular $n \times n$ matrix with all entries one. It is obvious that A is maximally monotone.

For the numerical comparison, we set the parameters for Algorithm 3.1: $\gamma = 0.1$, $\lambda_0 = 0.01$, $\mu = 0.9$, $\eta_k = \frac{1}{(k+1)^7}$ and $\delta_k = \frac{1}{(5k+2)^8}$.

For Algorithm 3.1 in [31], we set the parameters: $\gamma = 0.1$ and $\lambda_k = \frac{k}{(2k+1)L}$.

For Algorithm 1 in [10], we set the parameters: $\gamma = 0.1$, $\lambda_k = \frac{1}{2L}$, $\alpha = 0.3$, $\tau_k = \frac{1}{k^2}$, $\eta_k = \frac{1}{5k+1}$ and $\theta_k = 0.8 - \eta_k$.

Setting $\|x_{k+1} - x_k\| \leq \varepsilon$ as the stop criterion, we get the results with the number of iterations, CPU time and different ε in Tables 1 and 2. We can see from both Tables 1 and

TABLE 1. Comparison of Algorithm 3.1, Algorithm 3.1 in [31] and Algorithm 1 in [10] with $n = 500$

ε	Algorithm 3.1		Algorithm 3.1 in [31]		Algorithm 1 in [10]	
	CPU time (sec)	Iterations	CPU time (sec)	Iterations	CPU time (sec)	Iterations
10^{-2}	0.4228	69	0.4896	84	0.3767	70
10^{-5}	0.8747	135	1.5676	276	1.4867	277
10^{-10}	1.5604	244	3.3727	611	3.2118	589
10^{-15}	2.1503	353	5.1333	948	4.7967	905
10^{-20}	2.6121	463	7.0057	1283	6.6597	1220

2 that the number of iterations and CPU time of Algorithm 3.1 are less than Algorithm 3.1 in [31] and Algorithm 1 in [10]. This means that our Algorithm 3.1 performs better than other algorithms for each ε.

Next, we show the graphs of error plotting with the stop criterion $\varepsilon = 10^{-10}$.

TABLE 2. Comparison of Algorithm 3.1, Algorithm 3.1 in [31] and Algorithm 1 in [10] with $n = 700$

ε	Algorithm 3.1		Algorithm 3.1 in [31]		Algorithm 1 in [10]	
	CPU time (sec)	Iterations	CPU time (sec)	Iterations	CPU time (sec)	Iterations
10^{-2}	1.0009	75	1.1228	88	0.8879	75
10^{-5}	1.8836	139	3.4410	281	3.3517	283
10^{-10}	3.3561	248	7.6150	615	7.0599	594
10^{-15}	4.7721	358	11.4553	954	11.1163	912
10^{-20}	5.8551	467	17.0702	1290	15.5555	1229

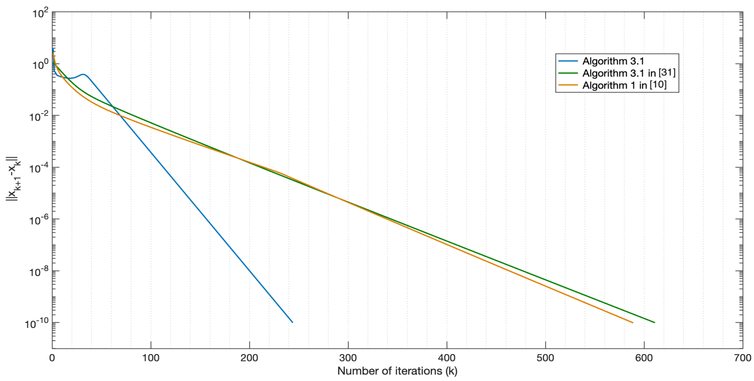

FIGURE 1. Plotting graph of comparison for each algorithm with $n = 500$

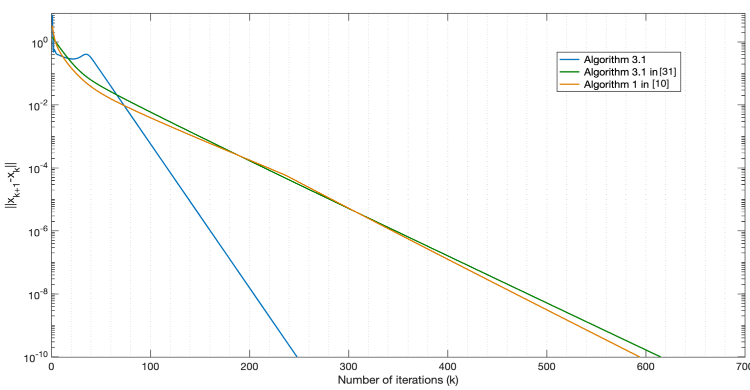

FIGURE 2. Plotting graph of comparison for each algorithm with $n = 700$

In Figures 1 and 2, we observe that Algorithm 3.1 has a better convergence than Algorithm 3.1 in [31] and Algorithm 1 in [10] in terms of iterations.

5. APPLICATIONS

5.1. **Convex minimization problem (CMP).** Next, we study the convex minimization problem (CMP):

$$\min_{x \in H}(g(x) + f(x)), \tag{5.48}$$

where H is a real Hilbert space, $g : H \to (-\infty, +\infty]$ is proper, lower semicontinuous and covex and $f : H \to \mathbb{R}$ is convex and differentiable with the Lipschitz continuous gradient denoted by ∇f. It is known that x^* is a minimizer of $g + f$ if and only if

$$0 \in (\partial g + \nabla f)(x^*), \tag{5.49}$$

where ∂g denotes the subdifferential of g.

In a real Hilbert space H, the proximal operator of g is defined by

$$Prox_{\beta g}(x) := \text{argmin}_{v \in H} \left\{ g(v) + \frac{1}{2\beta}\|v - x\|^2 \right\}, x \in H, \beta > 0.$$

It is well-known that

$$Prox_{\beta g}(x) = (I + \beta \partial g)^{-1}(x) = J_\beta^{\partial g}(x),$$

where ∂g is the subdifferential of g defined by

$$\partial g(x) := \{x^* \in H : g(x) + \langle y - x, x^* \rangle \le g(y),\ y \in H\}.$$

From [5], ∂g is a maximal monotone operator and $prox_{\beta g}$ is firmly nonexpansive.

So we obtain the following results.

Algorithm 5.1. *Suppose that* $\{\eta_k\}$ *and* $\{\delta_k\}$ *are nonnegative sequences satisfying* $\sum_{k=1}^{\infty} \eta_k < +\infty$ *and* $\sum_{k=1}^{\infty} \delta_k < +\infty$. *Let* $\gamma \in (0, 2)$, $\lambda_0 > 0$, $\mu \in (0, 1)$ *and* x_{-1}, x_0 *and* x_1 *be chosen arbitrary. Calculate* x_{k+1} *as follows:*

$$\begin{aligned} w_k &= x_k + \eta_k(x_k - x_{k-1}) + \delta_k(x_{k-1} - x_{k-2}) \\ y_k &= Prox_{\lambda_k g}(w_k - \lambda_k f(w_k)) \\ d(w_k, y_k) &= (w_k - y_k) - \lambda_k(f(w_k) - f(y_k)) \\ x_{k+1} &= w_k - \gamma \beta_k d(w_k, y_k), \end{aligned}$$

where

$$\lambda_{k+1} = \begin{cases} \min\{\frac{\mu\|w_k - y_k\|}{\|f(w_k) - f(y_k)\|}, \lambda_k\} & \text{if } f(w_k) - f(y_k) \neq 0 \\ \lambda_k & \text{otherwise} \end{cases}$$

and

$$\beta_k = \frac{\phi(w_k, y_k)}{\|d(w_k, y_k)\|^2},\ \phi(w_k, y_k) = \langle w_k - y_k, d(w_k, y_k) \rangle.$$

5.2. **Data classification problem.** In this section, we apply Algorithm 5.1 to data classification problem in heart failure prediction [35]. Heart failure refers to a wide range of symptoms caused by abnormalities in the functioning of the heart. It may develop due to abnormalities in either the structure or functioning of a patient's heart, which, as a result, causes the heart to become inefficient in pumping blood to the rest of the body or in retrieving blood from the rest of the body.

In particular, we apply extreme learning machine (ELM) to predict whether a patient is prone to heart failure depending on multiple attributes and compare results with provided by the machine learning algorithms. This dataset [35] involves 918 observations, 11 attributes and output class which are presented in Table 3.

TABLE 3. Details and statistical quantisation of all attributes

Attributes	Description	$\bar{x}$	S.D.	Max	Min	C.V.
Age	Age of the patient (years)	53.51	9.43	77	28	0.18
Sex	Sex of the patient	1.21	0.41	2	1	0.34
Chest Pain Type	Chest pain type	3.25	0.93	4	1	0.29
RestingBP	Resting blood pressure (mm Hg)	132.40	18.51	200	0	0.14
Cholesterol	Serum cholesterol (mm/dl)	198.80	109.38	603	0	0.55
FastingBS	Fasting blood sugar	0.23	0.42	1	0	1.81
RestingECG	Resting electrocardiogram results	1.60	0.81	3	1	0.50
MaxHR	Maximum heart rate achieved	136.81	25.46	202	60	0.19
Exercise Angina	Exercise-induced angina	1.60	0.50	2	1	0.31
Oldpeak	ST (Numeric value measured in depression)	0.89	1.07	6.2	-2.6	1.20
ST_Slope	The slope of the peak exercise ST segment	1.64	0.61	3	1	0.37
Output class	Heart disease, Normal	-	-	-	-	-

$\bar{x}$: Mean, S.D.: Standard deviation, C.V.: Coefficient of variation

Let $\{(x_k, y_k) : x_k \in \mathbb{R}^N, y_k \in \mathbb{R}^M, k = 1, 2, 3, ..., W\}$ be a training set consisting of W distinct samples and x_k, y_k are represented in Table 4. Given a single hidden layer of ELM, the output function at the i-th hidden node is defined as follows:

$$h_i(x) = U(\langle a_i, x\rangle + b_i),$$

where U, a_i and b_i are denoted in Table 4.

TABLE 4. Notations of parameters

Notations	Meaning
x_k	The input training data
y_k	The training target
a_i	The weight at the i-th hidden node
b_i	The bias of the i-th hidden node
ω_i	The optimal weight at the i-th hidden node to output layer
U	The activation function
L	The number of hidden nodes

The single-hidden layer feed forward neural networks (SLFNs) with L hidden nodes is defined as:

$$O_n = \sum_{i=1}^{L} \omega_i h_i(x_n),$$

where ω_i is defined in Table 4. The hidden layer output matrix A is defined as follows:

$$A = \begin{bmatrix} U(\langle a_1, x_1\rangle + b_1) & \cdots & U(\langle a_L, x_1\rangle + b_L) \\ \vdots & \ddots & \vdots \\ U(\langle a_1, x_W\rangle + b_1) & \cdots & U(\langle a_L, x_W\rangle + b_L) \end{bmatrix}$$

We aim to find an optimal weight $\omega = [\omega_1, ..., \omega_L]^T$ by ELM such that $A\omega = \chi$, where $\chi = [t_1, ..., t_W]^T$ is the training target data. On the otherhand, the convex minimization problem is to find the solution ω via the least absolute shrinkage and selection operator (LASSO) [29] as follows:

$$\min_{\omega \in \mathbb{R}^L} \{\|A\omega - \chi\|_2^2 + \xi\|\omega\|_1\}, \tag{5.50}$$

where ξ is a regularization parameter. We see that if $f(\omega) = \|A\omega - \chi\|_2^2$ and $g(\omega) = \xi\|\omega\|_1$, then the problem (5.50) is reduced to the problem (5.48).

We select the binary cross-entropy loss function in conjunction with the sigmoid activation function defined by

$$Loss = -\frac{1}{J}\sum_{j=1}^{J} v_j \log \hat{v}_j + (1 - v_j)\log(1 - \hat{v}_j) \tag{5.51}$$

where $\hat{v}_j$ and v_j are the j-th scalar value in the model output and the corresponding target value, respectively. The number of scalar values in the model output is defined by J.

Confusion matrix is defined as follows

TABLE 5. Confusion matrix for binary classification

		Prediction	
		Positive	Negative
Actual value	Positive	True positive (TP)	False negative (FN)
	Negative	False positive (FP)	True negative (TN)

Accuracy of algorithm is represented by accuracy, precision, recall and F1-score, which are calculated by Table 5:

- Precision = $\dfrac{\text{TP}}{\text{TP} + \text{FP}} \times 100\%$
- Recall = $\dfrac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%$
- Accuracy = $\dfrac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \times 100\%$
- F1-score = $\dfrac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$.

We use sigmoid for the activation function with hidden nods $L = 40$ and set regularization $\xi = 1 \times 10^{-5}$. For Algorithm 5.1, we set $x_{-1}, x_0, x_1 = (1, 1, ..., 1)$, $\gamma = 1.9$, $\lambda_0 = 1 \times 10^{-4}$, $\mu = 0.9$, $\eta_k = \frac{1}{(k+1)^7}$ and $\delta_k = \frac{1}{(k+100)^2}$. We compare Algorithm 5.1 with other traditional machine leaning method, the results are reported in Table 6.

TABLE 6. The performance in comparison of Algorithm 5.1 with traditional machine learning methods

Machine leaning method	Training time	Precision	Recall	F1-score	Accuracy
Logistic regression	7.4022	87.2	86.2	86.7	85.2
K neighbors	14.4760	82.9	83.0	82.9	81.2
Support vector machine	5.2009	88.2	87.0	87.6	86.2
Random forest	16.3060	90.2	86.3	88.2	86.6
Decision tree	5.3715	85.4	83.0	84.2	82.2
ELM (Algorithm 5.1)	0.1896	87.3	100	93.2	87.3

In Table 6, we observe that Algorithm 5.1 performs the best accuracy, F1-score and recall. This shows that our algorithm has the highest probability of classifying for heart failure prediction dataset [35].

Next, we show the results for loss value of training data and testing data in Table 7.

TABLE 7. Loss values of Algorithm 5.1

Iterations	Loss	
	Training	Test
1	0.284051	0.167517
2	1.321680	1.327856
3	0.250835	0.438841
4	0.246140	0.407872
⋮	⋮	⋮
698	0.213145	0.263093
699	0.213140	0.263091
700	0.213134	0.263089

We present graphs of the accuracy and loss of training data and testing data for overfitting of Algorithm 5.1.

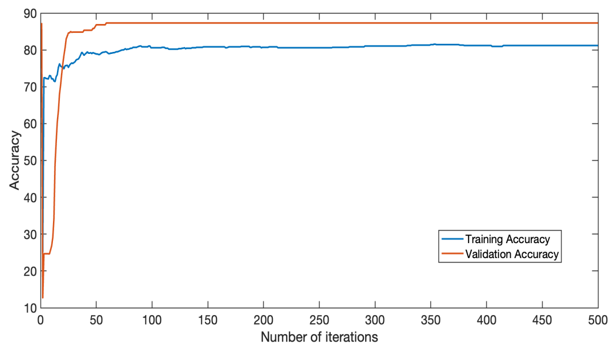

FIGURE 3. Plotting accuracy of Algorithm 5.1

FIGURE 4. Plotting loss of Algorithm 5.1

In Figures 3 and 4, we see that training accuracy (blue line) and validation accuracy (red line) increase. Moreover, the training loss and validation loss values have decreased. This means that Algorithm 5.1 can be used to classify effectively and has a good fitting model in the training dataset [35].

6. Conclusions

In this paper, we have introduced a contraction algorithm using two inertial terms with updated stepsize for solving the variational inclusion problem in Hilbert spaces. Under some suitable conditions, we have provided the weak convergence of the algorithm. The efficiency of our algorithm has been shown by comparing our algorithm with other algorithms in the literature review in finite dimensional spaces. Moreover, our algorithm has been applied to the data classification problem in heart failure prediction dataset [35]. The results of our algorithm to predict disease is 87.3% which is more efficient than traditional machine learning methods.

Acknowledgments
This research work was partially supported by Chiang Mai University and S. Kesornprom was supported by CMU Proactive Researcher, Chiang Mai University [grant number 789/2566]. The second author P. Cholamjiak was supported by University of Phayao and Thailand Science Research and Innovation Fund (Fundamental Fund 2024).

References

[1] Anh, P. K.; Thong, D. V.; Dung, V. T. A strongly convergent Mann-type inertial algorithm for solving split variational inclusion problems. *Optim. Eng.* **22** (2021), 159–185.
[2] Alvarez, F.; Attouch, H. An inertial proximal method for maximal monotone operators via discretization of a nonlinear oscillator with damping. *Set Valued Var. Anal.* **9** (2001), no. 1, 3–11.
[3] Brézis, H.; Chapitre, I. I. Operateurs Maximaux Monotones. *North-Holland Math. Stud.* **5** (1973), 19-51.
[4] Beck, A.; Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.* **2** (2009), 183–202.
[5] Butnariu, D.; Iusem, A. N. *Totally convex functions for fixed points computation and infinite dimensional optimization*. (**40**). Springer Science and Business Media, 2000.
[6] Byrne, C.; Censor, Y.; Gibali, A. The split common null point problem. *J. Nonlinear Convex Anal.* **13** (2011), 759–775.
[7] Cai, X.; Gu, G.; He, B. On the O (1/t) convergence rate of the projection and contraction methods for variational inequalities with Lipschitz continuous monotone operators. *Comput. Optim. Appl.* **57** (2014), no. 2, 339–363.
[8] Ceng, L. C.; Al-Homidan, S. Algorithms of common solutions for generalized mixed equilibria, variational inclusions, and constrained convex minimization. *Abstract Appl. Anal.* (2014), Hindawi.
[9] Chen, C.; Ma, S.; Yang, J. A general inertial proximal point algorithm for mixed variational inequality problem. *SIAM J. Optim.* **25** (2015), no. 4, 2120–2142.
[10] Dey, S. A hybrid inertial and contraction proximal point algorithm for monotone variational inclusions. *Numer. Alg.* **93** (2023), no. 1, 1–25.
[11] Dong, Q. L.; Cho, Y. J.; Rassias, T. M. General inertial Mann algorithms and their convergence analysis for nonexpansive mappings. *Appl. Non. Anal.* (2018), 175–191.
[12] Fang, Y. P.; Huang, N. J. H-monotone operator and resolvent operator technique for variational inclusions. *Appl. Math. Comput.* **145** (2003), no. (2-3) 795–803.
[13] Gabeleh, M., Shahzad, N. Existence and uniqueness of a solution for some nonlinear programming problems. *Mediterr. J. Math.* **12** (2015), 133–146.
[14] Goebel, K.; Kirk, W. A. *Topics in metric fixed point theory* (no. 28). Cambridge university press, 1990.
[15] Guo, C.; Zhao, J.; Dong, Q. L. A stochastic two-step inertial Bregman proximal alternating linearized minimization algorithm for nonconvex and nonsmooth problems. *Numer. Alg.* (2023), 1–50.
[16] Hanjing, A.; Suantai, S. A fast image restoration algorithm based on a fixed point and optimization method. *Mathematics* **8** (2020), no. 3, 378.
[17] Huang, N. J. A new completely general class of variational inclusions with noncompact valued mappings. *Comput. Math. Appl.* **35** (1998), no. 10, 9–14.
[18] Izuchukwu, C.; Shehu, Y.; Dong, Q. L. Two-step inertial forward-reflected-backward splitting based algorithm for nonconvex mixed variational inequalities. *J. Comput. Appl. Math.* **426** (2023), 115093.
[19] Iyiola, O. S.; Shehu, Y. Convergence results of two-step inertial proximal point algorithm. *Appl. Numer. Math.* **182** (2022), 57–75.
[20] Moudafi, A. Split monotone variational inclusions. *J. Optim. Theory Appl.* **150** (2011), no. 2, 275–283.

[21] Opial, Z. Weak convergence of the sequence of successive approximations for nonexpansive mappings. *Bull Am Math Soc.* **73** (1967), no. 4, 591–597.
[22] Osilike, M. O.; Aniagbosor, S. C.; Akuchu, B. G. Fixed points of asymptotically demicontractive mappings in arbitrary Banach spaces. *Panam. Math. J.* **12** (2002), no. 2, 77–88.
[23] Polyak, B. T. Some methods of speeding up the convergence of iteration methods. *USSR Comput. Math. Math. Phy.* **4** (1964), no. 5, 1–17.
[24] Qin, X.; Yao, J. C. A viscosity iterative method for a split feasibility problem. *J. Nonlinear Convex Anal.* **20** (2019), no. 8, 1497–1506.
[25] Rockafellar, R. T. Monotone operators and the proximal point algorithm. *SIAM J. Control Optim.* **14** (1976), no. 5, 877–898.
[26] Shehu, Y.; Yao, J. C. Weak convergence of two-step inertial iteration for countable family of quasi-nonexpansive mappings. *Annals Math. Sci. Appl.* **7** (2022), no. 2, 259–279.
[27] Solodov, M. V.; Svaiter, B. F. A hybrid projection-proximal point algorithm. *J. Convex Anal.* **6** (1999), no. 1, 59–70.
[28] Solodov, M. V.; Svaiter, B. F. A hybrid approximate extragradient-proximal point algorithm using the enlargement of a maximal monotone operator. *Set Valued Var. Anal.* **7**, no. 4, 323–345.
[29] Tibshirani, R. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Series B Stat. Methodol.* **58** (1996), 267–288.
[30] Hieu, D. V. ; Anh, P. K.; Muu, L. D. Modified forward-backward splitting method for variational inclusions. *4OR* **19** (2021), 127–151.
[31] Zhang, C.; Wang, Y. Proximal algorithm for solving monotone variational inclusion. *Optimization* **67** (2018), no. 8, 1197–1209.
[32] Zhang, C.; Dong, Q. L.; Chen, J. Multi-step inertial proximal contraction algorithms for monotone variational inclusion problems. *Carpathian J. Math.* **36** (2020), no. 1, 159–177.
[33] Zhao, J.; Dong, Q. L.; Rassias, M. T.; Wang, F. Two-step inertial Bregman alternating minimization algorithm for nonconvex and nonsmooth problems. *J. Glob. Optim.* **84** (2022), no. 4, 941–966.
[34] Zeng, L. C.; Guu, S. M.; Yao, J. C. Characterization of H-monotone operators with applications to variational inclusions. *Comput. Math. Appl.* **50** (2005), no. (3-4), 329-337.
[35] Fedesoriano. (September 2021). Heart Failure Prediction Dataset. Retrieved [Date Retrieved] from *https://www.kaggle.com/fedesoriano/heart-failure-prediction.*

[1]DEPARTMENT OF MATHEMATICS, FACULTY OF SCIENCE
CHIANG MAI UNIVERSITY
RESEARCH CENTER IN OPTIMIZATION AND COMPUTATIONAL INTELLIGENCE FOR BIG DATA PREDICTION,
CHIANG MAI, 50200, THAILAND
Email address: suthep.s@cmu.ac.th

[2]CHIANG MAI UNIVERSITY
OFFICE OF RESEARCH ADMINISTRATION
CHIANG MAI 50200, THAILAND
Email address: suparat.ke@gmail.com

[3]UNIVERSITY OF PHAYAO
SCHOOL OF SCIENCE
PHAYAO 56000, THAILAND
Email address: prasit.ch@up.ac.th

[3]UNIVERSITY OF PHAYAO
SCHOOL OF SCIENCE
PHAYAO 56000, THAILAND
Email address: papatsara.inkrong@gmail.com