CARPATHIAN J. MATH.
Volume **40** (2024), No. 2,
Pages 459 - 476

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.2024.02.15

Modified iterative schemes with two inertia and linesearch rule for split variational inclusion and applications to image deblurring and diabetes prediction

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

ABSTRACT. The purpose of this paper is to introduce a new inertial iterative algorithm involving the linesearch rule for the split variational inclusion in real Hilbert spaces. We use two inertial terms to accelerate the convergence speed of the proposed algorithm. Under suitable conditions, we prove its weak convergence theorem. Finally, in applications, we apply our algorithm to image deblurring and data classification problem in predicting diabetes mellitus.

1. INTRODUCTION

Let $\mathcal{H}$ be a real Hilbert space with the inner product $\langle\cdot,\cdot\rangle$ and the induced norm $\|\cdot\|$. Let $G:\mathcal{H}\to 2^{\mathcal{H}}$ be a set-valued maximal monotone mapping. The resolvent mapping $J_r^G:\mathcal{H}\to\mathcal{H}$ associated with G is defined by

$$J_r^G(x)=(I+rG)^{-1}(x),\ \forall x\in\mathcal{H},$$

for some $r>0$, where I is an identity operator on $\mathcal{H}$.

In this paper, we study the split variational inclusion problem (SVIP) which is formulated as follows:

$$\text{find a point } x^*\in\mathcal{H}_1 \text{ such that } 0\in G_1x^* \text{ and } 0\in G_2(Ax^*),$$

where $G_1:\mathcal{H}_1\to 2^{\mathcal{H}_1}$ and $G_2:\mathcal{H}_2\to 2^{\mathcal{H}_2}$ are multi-valued maximal monotone mappings, $\mathcal{H}_1$ and $\mathcal{H}_2$ are Hilbert spaces and $A:\mathcal{H}_1\to\mathcal{H}_2$ is a bounded linear operator. The split variational inclusion problem was studied by Moudafi [25] and Censor et al. [7]. The solution set of SVIP is denoted by

$$\Omega=\{x^*\in\mathcal{H}_1 : 0\in G_1x^* \text{ and } 0\in G_2(Ax^*)\}.$$

It is known that SVIP includes, as special cases, the split minimization problem, the split variational inequality problem, the split saddle point problem, the split equilibrium problem and the split feasibility problem; see, for instance, [3, 6, 17, 25, 31, 36]. In applications, SVIP can be reduced to image processing, signal processing and data science; see also [9, 10, 13, 26].

In recent years, numerous iterative algorithms for solving SVIP have been investigated by several authors [11, 20, 27]. A classical iterative algorithm for solving SVIP is originally introduced by Martinet [22] for solving convex minimization problem and Rockafellar [32] for a maximal monotone operator. Let $r_n\in(0,\infty)$ and $x_1\in\mathcal{H}_1$

$$x_{n+1}=J_{r_n}^{G_1}(x_n),$$

Received: 17.09.2023. In revised form: 12.03.2024. Accepted: 19.03.2024
2010 *Mathematics Subject Classification.* 49J40, 47J20, 35A15, 65K10, 97C70, 97U70.
Key words and phrases. *variational inclusion problem, linesearch rule, inertial term, weak convergence, data classification, diabetes mellitus.*
Corresponding author: Suparat Kesornprom; suparat.ke@gmail.com

where $r_n > 0$. This is called proximal algorithm. Byrne et al. [6] established the weak convergence of the following algorithm to solve SVIP: $x_1 \in \mathcal{H}_1$ and

$$x_{n+1} = J_{r_n}^{G_1}(x_n - r_n A^*(I - J_{r_n}^{G_2})Ax_n),$$

where $r_n \in (0, 2/\|A\|^2)$ and A^* denotes the adjoint operator of A from $\mathcal{H}_2$ to $\mathcal{H}_1$.

Alvarez and Attouch [2] studied the inertial proximal algorithm: $x_0, x_1 \in \mathcal{H}_1$ and

$$x_{n+1} = J_{r_n}^{G_1}(x_n + \theta_n(x_n - x_{n-1})),$$

where $\theta_n \in [0, 1)$ and $r_n \in (0, \infty)$. They established that if $\sum_{n=1}^{\infty} \theta_n \|x_n - x_{n-1}\|^2 < \infty$ holds, then the sequence $\{x_n\}$ converges weakly to an element of SVIP. The term $\theta_n(x_n - x_{n-1})$ represents the so-called inertia; see also [5, 30]. Several authors aim to construct various efficient iterative algorithms with inertial technique [1, 12, 21, 23, 33, 34].

Using inertial technique, Chuang [12] proposed the following hybrid inertial proximal algorithm: set $x_0, x_1 \in \mathcal{H}_1$, $\rho \in (0, 1)$ and $r_n > 0$, and define

$$\begin{aligned} u_n &= x_n + \theta_n(x_n - x_{n-1}), \\ y_n &= J_{r_n}^{G_1}(u_n - \lambda_n A^*(I - J_{r_n}^{G_2})Au_n), \\ x_{n+1} &= J_{r_n}^{G_1}(u_n - \gamma_n d(u_n, y_n)), \end{aligned}$$

where

$$\begin{aligned} d(u_n, y_n) &= u_n - y_n - \lambda_n(A^*(I - J_{r_n}^{G_2})Au_n - A^*(I - J_{r_n}^{G_2})Ay_n), \\ \gamma_n &= \frac{\langle u_n - y_n, d(u_n, y_n)\rangle}{\|d(u_n, y_n)\|^2}, \end{aligned}$$

and $\lambda_n > 0$ such that

$$\lambda_n \|A^*(I - J_{r_n}^{G_2})Au_n - A^*(I - J_{r_n}^{G_2})Ay_n\| \leq \rho\|u_n - y_n\|.$$

They proved that if $\{\theta_n\}$ is a nonnegative sequence satisfying $\sum_{n=1}^{\infty} \theta_n \|x_n - x_{n-1}\|^2 < \infty$, then the sequence $\{x_n\}$ converges weakly to an element of SVIP.

In 2018, Dong et al. [14] introduced the following algorithm

$$\begin{aligned} w_n &= x_n + \alpha_n(x_n - x_{n-1}), \\ y_n &= x_n + \delta_n(x_n - x_{n-1}), \\ x_{n+1} &= (1 - \lambda_n)w_n + \lambda_n U(y_n), \end{aligned}$$

where $\{\alpha_n\}$ and $\{\delta_n\}$ are nonnegative sequences and $U : \mathcal{H} \to \mathcal{H}$ is a nonexpansive mapping. They proved a weak convergence theorem for this algorithm.

Inspired by previous works, we construct new inertial iterative algorithms for SVIP. This is based on inertial technique and linesearch stepsize. Furthermore, we prove weak convergence theorem of the proposed algorithm under some conditions. Our result can be applied to solve the split feasibility problem which relates to data classification problem. Numerical experiments are supported to illustrate the advantages of our method.

The frame of the paper is as follows. In section 2, preliminaries and lemmas are presented for our analysis. In section 3, our algorithm is introduced and analyzed. Then, in section 4, we present numerical experiments to show the performance of our algorithm including a comparison with algorithms in literature. In Section 5, we show the application to image deblurring and data classification. Finally, in Section 6, we give conclusion.

2. Preliminaries and lemmas

In this section, we present fundamental definitions and lemmas. Let $\mathcal{H}$ be a real Hilbert space.

Recall that a mapping $U : \mathcal{H} \to \mathcal{H}$ is said to be
(1) nonexpansive if

$$\|Ux - Uy\| \leq \|x - y\|, \ \forall x, y \in \mathcal{H};$$

(2) firmly-nonexpansive if

$$\langle Ux - Uy, x - y\rangle \geq \|Ux - Uy\|^2, \ \forall x, y \in \mathcal{H}.$$

We know that if U is firmly-nonexpansive, then $I - U$ is also firmly-nonexpansive.
(3) L-Lipschitz continuous, if there exists a constant $L > 0$ such that

$$\|Ux - Uy\| \leq L\|x - y\|, \ \forall x, y \in \mathcal{H}.$$

The operator $G : \mathcal{H} \to 2^{\mathcal{H}}$ is called monotone if

$$\langle u - v, x - y\rangle \geq 0,$$

for all $(x, u), (y, v) \in \mathrm{graph}(G)$, where graph$(G)$ is defined by

$$\mathrm{graph}(G) := \{(x, u) \in \mathcal{H} \times \mathcal{H} : u \in G(x)\}.$$

The operator $G : \mathcal{H} \to 2^{\mathcal{H}}$ is maximally monotone if graph(G) is not properly contained in the graph of any other monotone operators.

Lemma 2.1. [16] *Let E be a nonempty closed convex subset of a real Hilbert space $\mathcal{H}$ and let $U : E \to E$ be a nonexpansive mapping. If $x_n \rightharpoonup x \in E$ and $\lim_{n\to\infty} \|x_n - Ux_n\| = 0$, then $x = Ux$.*

We define $G^{-1}(0) = \{x \in \mathcal{H} : 0 \in Gx\}$, $\mathcal{D}(U)$ is the domain of U, and $Fix(U)$ denotes the fixed point set of U, i.e. $Fix(U) = \{x \in \mathcal{H} : x = Ux\}$.

Lemma 2.2. [11, 24] *Let $\mathcal{H}$ be a real Hilbert space, $G : \mathcal{H} \to 2^{\mathcal{H}}$ be a set-valued maximal monotone mapping. Thus,*
(i) J_r^G is a single-valued and firmly nonexpansive mapping for each $r > 0$;
(ii) $\mathcal{D}(J_r^G) = \mathcal{H}$ and $Fix(J_r^G) = \{x \in \mathcal{D}(G) : 0 \in Gx\}$;
(iii) $\|x - J_r^G x\| \leq \|x - J_\gamma^G x\|$ for all $0 < r \leq \gamma$ and for all $x \in \mathcal{H}$;
(iv) Assume that $G^{-1}(0) \neq \emptyset$. Then $\|x - J_r^G x\|^2 + \|J_r^G - x^*\|^2 \leq \|x - x^*\|^2$ for each $x \in \mathcal{H}$, each $x^* \in G^{-1}(0)$, and each $r > 0$.
(v) Assume that $G^{-1}(0) \neq \emptyset$. Then $\langle x - J_r^G x, J_r^G x - w\rangle \geq 0$ for each $x \in \mathcal{H}$, each $w \in G^{-1}(0)$, and each $r > 0$.

Lemma 2.3. [11] *Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be real Hilbert spaces, $A : \mathcal{H}_1 \to \mathcal{H}_2$ be a bounded linear operator. Let $r > 0$, $\gamma > 0$, $G_1 : \mathcal{H}_1 \to 2^{\mathcal{H}_1}$ and $G_2 : \mathcal{H}_2 \to 2^{\mathcal{H}_2}$ be set-valued maximal monotone mappings. Given any $x^* \in \mathcal{H}_1$.*
(i) If x^* is a solution of (SVIP), then $J_r^{G_1}(x^* - \gamma A^*(I - J_r^{G_2})Ax^*) = x^*$.
(ii) Suppose that $J_r^{G_1}(x^* - \gamma A^*(I - J_r^{G_2})Ax^*) = x^*$ and the solution set of (SVIP) is nonempty. Then x^* is a solution of (SVIP).

Lemma 2.4. [11] *Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be real Hilbert spaces, $A : \mathcal{H}_1 \to \mathcal{H}_2$ be a bounded linear operator and $r > 0$. Let $G : \mathcal{H}_2 \to 2^{\mathcal{H}_2}$ be a set-valued maximal monotone mapping. Define a mapping $U : \mathcal{H}_1 \to \mathcal{H}_1$ by $Ux := A^*(I - J_r^G)Ax$ for each $x \in \mathcal{H}_1$. Then*
(i) $\|(I - J_r^G)Ax - (I - J_r^G)Ay\|^2 \leq \langle Ux - Uy, x - y\rangle$ for all $x, y \in \mathcal{H}_1$;
(ii) $\|A^*(I - J_r^G)Ax - A^*(I - J_r^G)Ay\|^2 \leq \|A\|^2 \cdot \langle Ux - Uy, x - y\rangle$ for all $x, y \in \mathcal{H}_1$.

Lemma 2.5. [29] *Let $\{\Phi_n\}$, $\{\Psi_n\}$ and $\{\Theta_n\}$ be real positive sequences such that*

$$\Phi_{n+1} \leq (1+\Theta_n)\Phi_n + \Psi_n,\ n \geq 1.$$

If $\Sigma_{n=1}^{\infty}\Theta_n < +\infty$ and $\Sigma_{n=1}^{\infty}\Psi_n < +\infty$, then $\lim\limits_{n\to+\infty}\Phi_n$ exists.

Lemma 2.6. *(Opial theorem* [28]*) Let E be a nonempty subset of a real Hilbert space $\mathcal{H}$ and $\{x_n\}$ be a sequence in $\mathcal{H}$ that satisfies the following properties:*
(i) $\lim\limits_{n\to\infty}\|x_n - x\|$ exists for every $x \in E$;
(ii) each weak limit point of $\{x_n\}$ is in E.
Then $\{x_n\}$ converges weakly to a point in E.

3. Main results

Given $\mathcal{H}_1$ and $\mathcal{H}_2$ be real Hilbert spaces. We define $A : \mathcal{H}_1 \to \mathcal{H}_2$ as a linear and bounded operator and A^* is the adjoint operator of A. Let $G_1 : \mathcal{H}_1 \to 2^{\mathcal{H}_1}$ and $G_2 : \mathcal{H}_2 \to 2^{\mathcal{H}_2}$ be maximal monotone operators. We denote the solution set of SVIP by Ω, where Ω is nonempty.

Lemma 3.7. *Let $\varphi_{-1} \geq 0$, $\varphi_0 \geq 0$, $\{\varphi_n\}$, $\{\alpha_n\}$ and $\{\delta_n\}$ be nonnegative real sequences satisfying*

$$\varphi_{n+1} \leq (1+\alpha_n)\varphi_n + (\alpha_n+\delta_n)\varphi_{n-1} + \delta_n\varphi_{n-2},\ n \in \mathbb{N}.$$

Then

$$\varphi_{n+1} \leq K \cdot \prod_{j=1}^{n}(1+2\alpha_j+2\delta_j),\ n \in \mathbb{N}$$

where $K = \max\{\varphi_{-1}, \varphi_0, \varphi_1\}$. Furthermore, if $\sum_{n=1}^{\infty}\alpha_n < +\infty$ and $\sum_{n=1}^{\infty}\delta_n < +\infty$, then $\{\varphi_n\}$ is bounded.

Proof. By mathematical induction. (See also [19]). □

Algorithm 3.1. *Given $\gamma > 0$, $\ell \in (0,1)$ and $\mu \in (0,1)$. Let $\{r_n\}$, $\{\alpha_n\}$ and $\{\delta_n\}$ be nonnegative real sequences. Let $x_{-1}, x_0, x_1 \in H_1$ be arbitrary. For $n \geq 1$, calculate x_{n+1} as follows:*

$$w_n = x_n + \alpha_n(x_n - x_{n-1}) + \delta_n(x_{n-1} - x_{n-2})$$

$$y_n = J_{r_n}^{G_1}(w_n - \lambda_n A^*(I - J_{r_n}^{G_2})Aw_n) \tag{3.1}$$

$$x_{n+1} = J_{r_n}^{G_1}(w_n - \lambda_n A^*(I - J_{r_n}^{G_2})Ay_n) \tag{3.2}$$

where $\lambda_n = \gamma\ell^{m_n}$ and m_n is the smallest nonnegative integer satisfying the following:

$$\begin{aligned} & \lambda_n\langle A^*(I - J_{r_n}^{G_2})Aw_n - A^*(I - J_{r_n}^{G_2})Ay_n, x_{n+1} - y_n\rangle \\ \leq\ & \frac{\mu}{2}(\|w_n - y_n\|^2 + \|x_{n+1} - y_n\|^2). \end{aligned} \tag{3.3}$$

Lemma 3.8. *The linesearch rule (3.3) is well-defined and $\min\{\gamma, \frac{\mu\ell}{L}\} \leq \lambda_n \leq \gamma$.*

Proof. Obviously, (3.3) holds for all $0 < \lambda_n \leq \frac{\mu}{L}$. It is easy to see that $\lambda_n \leq \gamma$. Next, we will show that $\lambda_n \geq \min\{\gamma, \frac{\mu\ell}{L}\}$.
If $\lambda_n = \gamma$, then this lemma is proved.

If $\lambda_n < \gamma$, from the linesearch (3.3), we know that

$$\begin{aligned} & \mu(\|w_n - y_n\|^2 + \|x_{n+1} - y_n\|^2) \\ < \; & \frac{2\lambda_n}{\ell}\langle A^*(I - J_{r_n}^{G_2})Aw_n - A^*(I - J_{r_n}^{G_2})Ay_n, x_{n+1} - y_n\rangle \\ \leq \; & \frac{2\lambda_n}{\ell}\|A^*(I - J_{r_n}^{G_2})Aw_n - A^*(I - J_{r_n}^{G_2})Ay_n\|\|x_{n+1} - y_n\| \\ \leq \; & \frac{2L\lambda_n}{\ell}(\|w_n - y_n\|\|x_{n+1} - y_n\|) \\ \leq \; & \frac{\lambda_n}{\ell}L(\|w_n - y_n\|^2 + \|x_{n+1} - y_n\|^2), \end{aligned}$$

where $L = \|A\|^2$, which implies that $\lambda_n > \frac{\mu\ell}{L}$. Hence $\lambda_n \geq \min\{\gamma, \frac{\mu\ell}{L}\}$. □

Theorem 3.1. *Let $\{x_n\}$ be a sequence defined by Algorithm 3.1. Suppose that $\{r_n\}$ is a sequence in $[r, \infty)$ for some $r > 0$. If $\sum_{n=1}^{\infty} \alpha_n < +\infty$ and $\sum_{n=1}^{\infty} \delta_n < +\infty$, then $\{x_n\}$ converges weakly to a solution in Ω.*

Proof. Let $z \in \Omega$. Then $z \in G_1^{-1}(0)$ and $Az \in G_2^{-1}(0)$. From Lemma 2.4(i), we obtain

$$\begin{aligned} & \langle A^*(I - J_{r_n}^{G_2})Ay_n, y_n - z\rangle \\ = \; & \langle A^*(I - J_{r_n}^{G_2})Ay_n - A^*(I - J_{r_n}^{G_2})Az, y_n - z\rangle \\ = \; & \langle (I - J_{r_n}^{G_2})Ay_n - (I - J_{r_n}^{G_2})Az, Ay_n - Az\rangle \\ \geq \; & \|(I - J_{r_n}^{G_2})Ay_n\|^2. \end{aligned} \tag{3.4}$$

From (3.1), (3.2) and Lemma 2.2 (v), we obtain

$$\langle y_n - w_n + \lambda_n A^*(I - J_{r_n}^{G_2})Aw_n, x_{n+1} - y_n\rangle \geq 0. \tag{3.5}$$

From Lemma 2.2 (iv), we see that

$$\begin{aligned}
& \|x_{n+1}-z\|^2 \\
= \; & \|J_{r_n}^{G_1}(w_n-\lambda_n A^*(I-J_{r_n}^{G_2})Ay_n)-z\|^2 \\
\leq \; & \|w_n-\lambda_n A^*(I-J_{r_n}^{G_2})Ay_n-z\|^2-\|x_{n+1}-w_n+\lambda_n A^*(I-J_{r_n}^{G_2})Ay_n\|^2 \\
= \; & \|w_n-z\|^2-2\lambda_n\langle A^*(I-J_{r_n}^{G_2})Ay_n, w_n-z\rangle+\|\lambda_n A^*(I-J_{r_n}^{G_2})Ay_n\|^2 \\
& -\|x_{n+1}-w_n\|^2-2\lambda_n\langle A^*(I-J_{r_n}^{G_2})Ay_n, x_{n+1}-w_n\rangle-\|\lambda_n A^*(I-J_{r_n}^{G_2})Ay_n\|^2 \\
= \; & \|w_n-z\|^2-\|x_{n+1}-w_n\|^2-2\lambda_n\langle A^*(I-J_{r_n}^{G_2})Ay_n, x_{n+1}-z\rangle \\
= \; & \|w_n-z\|^2-\|x_{n+1}-w_n\|^2-2\lambda_n\langle A^*(I-J_{r_n}^{G_2})Ay_n, x_{n+1}-y_n\rangle \\
& -2\lambda_n\langle A^*(I-J_{r_n}^{G_2})Ay_n, y_n-z\rangle \\
= \; & \|w_n-z\|^2-\|x_{n+1}-y_n+y_n-w_n\|^2 \\
& -2\lambda_n\langle A^*(I-J_{r_n}^{G_2})Ay_n-A^*(I-J_{r_n}^{G_2})Aw_n+A^*(I-J_{r_n}^{G_2})Aw_n, x_{n+1}-y_n\rangle \\
& -2\lambda_n\langle A^*(I-J_{r_n}^{G_2})Ay_n, y_n-z\rangle \\
= \; & \|w_n-z\|^2-\|x_{n+1}-y_n\|^2-\|y_n-w_n\|^2-2\langle y_n-w_n, x_{n+1}-y_n\rangle \\
& -2\lambda_n\langle A^*(I-J_{r_n}^{G_2})Ay_n-A^*(I-J_{r_n}^{G_2})Aw_n, x_{n+1}-y_n\rangle \\
& -2\lambda_n\langle A^*(I-J_{r_n}^{G_2})Aw_n, x_{n+1}-y_n\rangle-2\lambda_n\langle A^*(I-J_{r_n}^{G_2})Ay_n, y_n-z\rangle \\
= \; & \|w_n-z\|^2-\|x_{n+1}-y_n\|^2-\|y_n-w_n\|^2 \\
& -2\langle y_n-w_n+\lambda_n A^*(I-J_{r_n}^{G_2})Aw_n, x_{n+1}-y_n\rangle \\
& +2\lambda_n\langle A^*(I-J_{r_n}^{G_2})Aw_n-A^*(I-J_{r_n}^{G_2})Ay_n, x_{n+1}-y_n\rangle \\
& -2\lambda_n\langle A^*(I-J_{r_n}^{G_2})Ay_n, y_n-z\rangle.
\end{aligned}$$

From (3.3), (3.4) and (3.5), we have

$$\begin{aligned}
\|x_{n+1}-z\|^2 \leq \; & \|w_n-z\|^2-\|x_{n+1}-y_n\|^2-\|y_n-w_n\|^2+\mu\|x_{n+1}-y_n\|^2 \\
& +\mu\|w_n-y_n\|^2-2\lambda_n\|(I-J_{r_n}^{G_2})Ay_n\|^2 \\
= \; & \|w_n-z\|^2-(1-\mu)\|x_{n+1}-y_n\|^2-(1-\mu)\|w_n-y_n\|^2 \\
& -2\lambda_n\|(I-J_{r_n}^{G_2})Ay_n\|^2. \qquad (3.6)
\end{aligned}$$

From definition of w_n, we see that

$$\begin{aligned}
\|w_n-z\| = \; & \|x_n+\alpha_n(x_n-x_{n-1})+\delta_n(x_{n-1}-x_{n-2})-z\| \\
\leq \; & \|x_n-z\|+\alpha_n\|x_n-x_{n-1}\|+\delta_n\|x_{n-1}-x_{n-2}\|. \qquad (3.7)
\end{aligned}$$

Since $\mu \in (0,1)$ and from (3.6) and (3.7), we see that

$$\begin{aligned}
\|x_{n+1}-z\| \leq \; & \|w_n-z\| \\
\leq \; & \|x_n-z\|+\alpha_n\|x_n-x_{n-1}\|+\delta_n\|x_{n-1}-x_{n-2}\| \\
\leq \; & \|x_n-z\|+\alpha_n(\|x_n-z\|+\|x_{n-1}-z\|)+\delta_n(\|x_{n-1}-z\|+\|x_{n-2}-z\|) \\
= \; & (1+\alpha_n)\|x_n-z\|+(\alpha_n+\delta_n)\|x_{n-1}-z\|+\delta_n\|x_{n-2}-z\|. \qquad (3.8)
\end{aligned}$$

From Lemma 3.7, we obtain

$$\|x_{n+1}-z\| \leq K\prod_{j=1}^{n}(1+2\alpha_j+2\delta_j),$$

where $K=\max\{\|x_1-z\|, \|x_0-z\|, \|x_{-1}-z\|\}$. Moreover, we obtain $\{x_n\}$ is bounded. Also, we have $\sum_{n=1}^{\infty}\alpha_n\|x_n-x_{n-1}\|<+\infty$ and $\sum_{n=1}^{\infty}\delta_n\|x_{n-1}-x_{n-2}\|<+\infty$. Using

Lemma 2.5 and (3.8), we have $\lim_{n\to\infty}\|x_n - z\|$ exists. On the otherhand, we get

$$\begin{aligned}
&\|w_n - z\|^2\\
= \ & \|x_n + \alpha_n(x_n - x_{n-1}) + \delta_n(x_{n-1} - x_{n-2}) - z\|^2\\
= \ & \|(x_n - z) + \alpha_n(x_n - x_{n-1})\|^2 + \delta_n^2\|x_{n-1} - x_{n-2}\|^2\\
&+2\langle x_n - z + \alpha_n(x_n - x_{n-1}), \delta_n(x_{n-1} - x_{n-2})\rangle\\
= \ & \|x_n - z\|^2 + \alpha_n^2\|x_n - x_{n-1}\|^2 + 2\langle x_n - z, \alpha_n(x_n - x_{n-1})\rangle + \delta_n^2\|x_{n-1} - x_{n-2}\|^2\\
&+2\langle x_n - z, \delta_n(x_{n-1} - x_{n-2})\rangle + 2\langle \alpha_n(x_n - x_{n-1}), \delta_n(x_{n-1} - x_{n-2})\rangle\\
\leq \ & \|x_n - z\|^2 + \alpha_n^2\|x_n - x_{n-1}\|^2 + 2\alpha_n\|x_n - z\|\|x_n - x_{n-1}\| + \delta_n^2\|x_{n-1} - x_{n-2}\|^2\\
&+2\delta_n\|x_n - z\|\|x_{n-1} - x_{n-2}\| + 2\alpha_n\delta_n\|x_n - x_{n-1}\|\|x_{n-1} - x_{n-2}\|. \qquad (3.9)
\end{aligned}$$

Substituting (3.9) into (3.6), we have

$$\begin{aligned}
&\|x_{n+1} - z\|^2\\
\leq \ & \|x_n - z\|^2 + \alpha_n^2\|x_n - x_{n-1}\|^2 + 2\alpha_n\|x_n - z\|\|x_n - x_{n-1}\| + \delta_n^2\|x_{n-1} - x_{n-2}\|^2\\
&+2\delta_n\|x_n - z\|\|x_{n-1} - x_{n-2}\| + 2\alpha_n\delta_n\|x_n - x_{n-1}\|\|x_{n-1} - x_{n-2}\|\\
&-(1-\mu)\|x_{n+1} - y_n\|^2 - (1-\mu)\|w_n - y_n\|^2 - 2\lambda_n\|(I - J_{r_n}^{G_2})Ay_n\|^2. \qquad (3.10)
\end{aligned}$$

Since $\sum_{n=1}^{\infty}\alpha_n\|x_n - x_{n-1}\| < +\infty$, $\sum_{n=1}^{\infty}\delta_n\|x_{n-1} - x_{n-2}\| < +\infty$ and $\lim_{n\to\infty}\|x_n - z\|$ exists, by (3.10), we obtain

$$\lim_{n\to\infty}\|x_{n+1} - y_n\| = 0$$

and

$$\lim_{n\to\infty}\|w_n - y_n\| = 0. \qquad (3.11)$$

We observe that

$$\begin{aligned}
\|w_n - x_n\| &= \|x_n + \alpha_n(x_n - x_{n-1}) + \delta_n(x_{n-1} - x_{n-2}) - x_n\|\\
&\leq \alpha_n\|x_n - x_{n-1}\| + \delta_n\|x_{n-1} - x_{n-2}\|\\
&\to 0. \qquad (3.12)
\end{aligned}$$

From (3.11) and (3.12), we have

$$\begin{aligned}
\|x_n - y_n\| &\leq \|x_n - w_n\| + \|w_n - y_n\|\\
&\to 0. \qquad (3.13)
\end{aligned}$$

Again from (3.10) and Lemma 3.8, we obtain

$$\lim_{n\to\infty}\|Ay_n - J_{r_n}^{G_2}Ay_n\| = 0. \qquad (3.14)$$

Using Lemma 2.2 (iii), we have

$$\lim_{n\to\infty}\|Ay_n - J_{r}^{G_2}Ay_n\| \leq \lim_{n\to\infty}\|Ay_n - J_{r_n}^{G_2}Ay_n\| = 0.$$

From (3.13) and (3.14), we have

$$\begin{aligned}
\|Ax_n - J_{r_n}^{G_2}Ax_n\| &\leq \|Ax_n - J_{r_n}^{G_2}Ax_n - Ay_n + J_{r_n}^{G_2}Ay_n\| + \|Ay_n - J_{r_n}^{G_2}Ay_n\|\\
&\leq 2\|A\|\|x_n - y_n\| + \|Ay_n - J_{r_n}^{G_2}Ay_n\|\\
&\to 0.
\end{aligned}$$

By Lemma 2.2 (iii), we also have

$$\lim_{n\to\infty}\|Ax_n - J_{r}^{G_2}Ax_n\| \leq \lim_{n\to\infty}\|Ax_n - J_{r_n}^{G_2}Ax_n\| = 0. \qquad (3.15)$$

Since $J_{r_n}^{G_1}$ is nonexpansive and from (3.11), (3.14), we have

$$\begin{aligned}\|y_n - J_{r_n}^{G_1} y_n\| &= \|J_{r_n}^{G_1}(w_n - \lambda_n A^*(I - J_{r_n}^{G_2})Ay_n) - J_{r_n}^{G_1} y_n\| \\ &\leq \|w_n - \lambda_n A^*(I - J_{r_n}^{G_2})Ay_n - y_n\| \\ &\leq \|w_n - y_n\| + \lambda_n \|A^*(I - J_{r_n}^{G_2})Ay_n\| \\ &\leq \|w_n - y_n\| + \lambda_n \|A\| \|(I - J_{r_n}^{G_2})Ay_n\| \\ &\to 0.\end{aligned}$$

By Lemma 2.2 (iii), we have

$$\lim_{n\to\infty} \|y_n - J_r^{G_1} y_n\| \leq \lim_{n\to\infty} \|y_n - J_{r_n}^{G_1} y_n\| = 0. \tag{3.16}$$

From (3.13) and (3.16), we have

$$\begin{aligned}\|x_n - J_r^{G_1} x_n\| &\leq \|x_n - y_n\| + \|y_n - J_r^{G_1} y_n\| + \|J_r^{G_1} y_n - J_r^{G_1} x_n\| \\ &\leq 2\|x_n - y_n\| + \|y_n - J_r^{G_1} y_n\| \\ &\to 0. \end{aligned} \tag{3.17}$$

Since $\{x_n\}$ is bounded, then there exists a subsequence $\{x_{n_i}\}$ of $\{x_n\}$ and $z^* \in \mathcal{H}$ such that $x_{n_i} \rightharpoonup z^*$. As A is a bounded linear operator, we have $Ax_{n_i} \rightharpoonup Az^*$. Using (3.15), (3.17), Lemma 2.1 and Lemma 2.2 (ii), we can deduce that $z^* \in \Omega$. By Lemma 2.6, we can conclude that the sequence $\{x_n\}$ converges weakly to a point in Ω. The proof is finished. □

4. Numerical illustrations

In this section, we provide a numerical example in finite dimensional spaces to compare convergence behavior of our proposed Algorithm 3.1 with Algorithm of Byrne et al. [6] and Algorithm of Chuang [12]. Let $\mathcal{H}_1 = \mathcal{H}_2 = \mathbb{R}^3$ and define the operators A, G_1 and G_2, respectively, as follows:

$$A = \begin{pmatrix} 6 & 3 & 1 \\ 8 & 7 & 5 \\ 3 & 6 & 2 \end{pmatrix}, G_1 = \begin{pmatrix} 6 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 3 \end{pmatrix}, G_2 = \begin{pmatrix} 7 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 2 \end{pmatrix}.$$

We choose the parameters in Algorithm 3.1 by $\gamma = 2$, $\ell = 0.1$, $\mu = 0.95$, $\alpha_n = \frac{1}{(20n+100)^7}$, $r_n = 1$ and $\delta_n = \frac{1}{(n+100)^7}$.

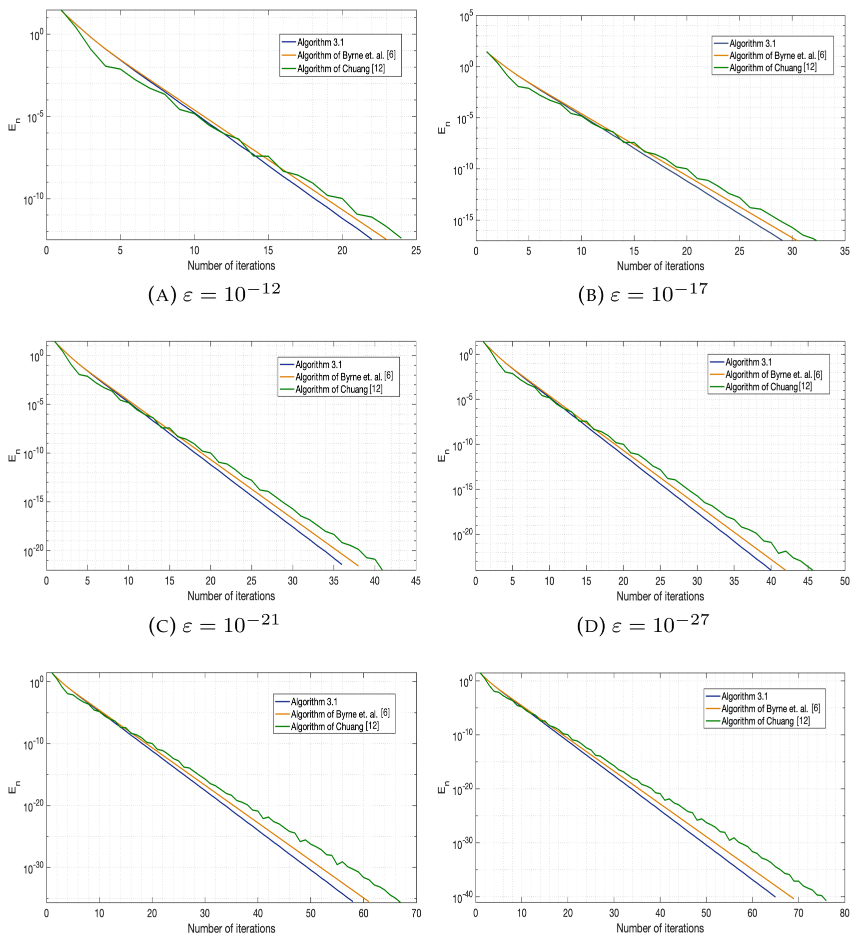
In Algorithm of Byrne et al. [6], we set $\lambda = \frac{0.1}{\|A\|^2}$.

In Algorithm of Chuang [12], we set $\theta = 0.95$, $\lambda_n = \frac{\rho}{\|A\|^2}$, $\rho = 0.95$ and $\theta_n = \begin{cases} \min\{\frac{1}{n^2\|x_n - x_{n-1}\|^2}, \theta\} & \text{if } x_n \neq x_{n-1}, \\ \theta & \text{otherwise.} \end{cases}$. We choose $x_{-1} = (-2, -14, 6)$, $x_0 = (13, -12, 25)$ and $x_1 = (-30, -3, 8)$. We use the stopping condition $E_n = \|x_{n+1} - x_n\| < \varepsilon$. Then, we test the convergence behavior with different ε. The outcomes are reported in Table 1 and Figure 2.

TABLE 1. The number of iterations and CPU time with different ε

ε	Byrne et al. [6]'s Algorithm		Chuang [12]'s Algorithm		Algorithm 3.1	
	Iteration	CPU time	Iteration	CPU time	Iteration	CPU time
10^{-12}	23	0.0080	24	0.0013	22	0.0925
10^{-17}	31	0.0183	33	0.0016	30	0.0082
10^{-21}	38	0.0212	41	0.0015	36	0.0087
10^{-27}	47	0.0038	52	0.0014	45	0.0058
10^{-30}	61	0.0027	67	0.0019	58	0.0079
10^{-40}	69	0.0035	76	0.0032	65	0.0186

Next, we demonstrate the numerical results with various ε values generated by Algorithm 3.1, Byrne et al. [6]'s Algorithm and Chuang [12]'s Algorithm as follows:



(A) $\varepsilon = 10^{-12}$ (B) $\varepsilon = 10^{-17}$

(C) $\varepsilon = 10^{-21}$ (D) $\varepsilon = 10^{-27}$

(E) $\varepsilon = 10^{-30}$ (F) $\varepsilon = 10^{-40}$

FIGURE 1. Errors of all algorithms with different ε

From Table 2 and Figure 2, we observe that our Algorithm 3.1 has less iterations than Byrne et al. [6]'s Algorithm and Chuang [12]'s Algorithm.

5. Applications

In this section, we present applications in image deblurring and data classification. In the first part, we consider special cases of the split variation inclusion problem to the split feasibility problem. The second part presents image deblurring and the third part presents the diabetes prediction in data classification problems. All numerical experiments were conducted on MacBook Pro M1 with ram 8 GB and were implemented in Matlab R2022b.

5.1. **Application to the split feasibility problem.** Consider the following split feasibility problem (SFP) suggested by Censor and Elfving [8].

$$\text{find } x^* \in C \text{ such that } Ax^* \in Q, \tag{5.18}$$

where C and Q are closed convex subsets of real Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively, and $A : \mathcal{H}_1 \to \mathcal{H}_2$ is a bounded linear operator.

The definition of the subdifferential ∂g for a proper lower semicontinuous convex function $g : \mathcal{H} \to (-\infty, \infty)$ is defined by

$$\partial g(x) = \{z \in \mathcal{H} : g(x) - g(y) \leq \langle z, x - y\rangle \ \forall y \in \mathcal{H}\},$$

for all $x \in \mathcal{H}$. Now, let C be a nonempty closed convex subset of $\mathcal{H}$, and let i_C be the indicator function of C which is defined as follows:

$$i_C(x) = \begin{cases} 0 & \text{if } x \in C, \\ \infty & \text{if } x \notin C. \end{cases}$$

Further, the normal cone $N_C u$ of C at $u \in C$ is defined as

$$N_C u = \{z \in \mathcal{H} : \langle z, v - u\rangle \leq 0, \ \forall v \in C\}.$$

It is known that i_C is a proper, lower semicontinuous and convex function on $\mathcal{H}$, hence, the subdifferential ∂i_C is a maximal monotone operator. Thus, the resolvent $J_r^{\partial i_C}$ for each $r > 0$ can be defined as

$$J_r^{\partial i_C}(x) = (I + r\partial i_C)^{-1}(x)$$

for all $x \in \mathcal{H}$.

It is known that for any $x \in C$, $\partial i_C x = N_C x$. Putting $y = J_r^{\partial i_C} x = (I + r\partial i_C)^{-1}(x)$, $r > 0$, we obtain

$$\begin{aligned} x \in y + r\partial i_C(y) \quad &\Leftrightarrow \quad x - y \in r\partial i_C y \\ &\Leftrightarrow \quad \langle x - y, z - y\rangle \leq 0 \, \forall z \in C \\ &\Leftrightarrow \quad y = P_C x. \end{aligned}$$

Consequently, we have the following results:

Algorithm 5.1. *Given $\gamma > 0$, $\ell \in (0,1)$ and $\mu \in (0,1)$. Let $\{\lambda_n\}$, $\{\alpha_n\}$ and $\{\delta_n\}$ be nonnegative real sequences. Let $x_{-1}, x_0, x_1 \in \mathcal{H}_1$ be arbitrary. For $n \geq 1$, compute x_{n+1} by*

$$\begin{aligned} w_n &= x_n + \alpha_n(x_n - x_{n-1}) + \delta_n(x_{n-1} - x_{n-2}) \\ y_n &= P_C(w_n - \lambda_n A^*(I - P_Q)Aw_n) \\ x_{n+1} &= P_C(w_n - \lambda_n A^*(I - P_Q)Ay_n) \end{aligned}$$

where $\lambda_n = \gamma\ell^{m_n}$ and m_n is the smallest nonnegative integer satisfying the following:

$$\begin{aligned} &\lambda_n\langle A^*(I - P_Q)Aw_n - A^*(I - P_Q)Ay_n, x_{n+1} - y_n\rangle \\ \leq \ &\frac{\mu}{2}(\|w_n - y_n\|^2 + \|x_{n+1} - y_n\|^2). \end{aligned}$$

Theorem 5.2. *Let $\{x_n\}$ be a sequence defined by Algorithm 5.1. If $\sum_{n=1}^{\infty} \alpha_n < +\infty$ and $\sum_{n=1}^{\infty} \delta_n < +\infty$, then $\{x_n\}$ converges weakly to a solution in (5.18).*

5.2. **Application to image deblurring.** In this section, we present numerical experiments to an image debluring. Let $C = [0, 255]^D$ with $D = M \times N$ where M is the pixels of width and N is the pixels of height of color image. Consider the minimization problem:

$$\min_{x \in C} \|Ax - y\|_2. \tag{5.19}$$

Therefore, the problem (5.18) can be reduced to (5.19) when $Q = \{y\}$ and $C = [0, 255]^D$. We will compare the following methods with $x_{-1} = (1, 1, 1, ..., 1) \in \mathbb{R}^N$, $x_0 = (0, 0, 0, ..., 0) \in \mathbb{R}^N$ and $x_1 = (1, 1, 1, ..., 1) \in \mathbb{R}^N$. We set parameters by

Algorithm of Byrne et al. [6]: $\lambda = \frac{0.3}{\|A\|^2}$;

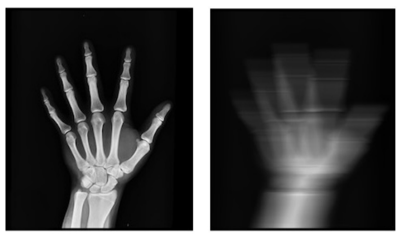
Algorithm of Chuang [12]: $\theta_n = \frac{1}{n^{1.1}\|x_n - x_{n-1}\|^2}$ and $\rho = 0.9$;

Algorithm of Vinh et al. [37]: $\alpha_n = \frac{1}{n^{1.1}\|x_n - x_{n-1}\|^2}$ and $\rho_n = 0.03$;

Algorithm of Dong et al. [15]: $\theta = 0.8$, $\gamma = 1$ and $\alpha = 0.2$;

Algorithm 5.1: $\gamma = 0.6$, $\ell = 0.8$, $\mu = 0.9$, $\alpha_n = \frac{1}{(20n+100)^7}$ and $\delta_n = \frac{1}{(n+100)^7}$.

We consider motion blur with motion length of 45 pixels and motion orientation $180°$, the image size 273×227 for RGB images.



(A) (B)

FIGURE 2. (A) The original image and (B) Motion blurred image

To measure the restored images, we use the peak-signal-to-noise ratio (PSNR) [35] defined by

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \tag{5.20}$$

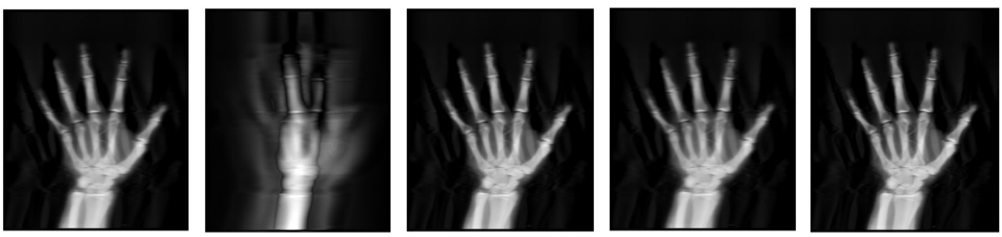
where MSE$= \|x_n - x\|^2$ with x is an original image. We also use structural similarity index measure (SSIM) [38] for measuring the similarity between two images. From definitions, it is clear that the high PSNR and SSIM values show the quality of restored images. We obtain numerical results as follows:

TABLE 2. The comparison of PSNR, SSIM and CPU time of the restored images

Algorithms	PSNR	SSIM	CPU time
[6, Algorithm 3.1]	24.8103	0.7190	12.0814
[12, Algorithm 3.1]	18.3787	0.5721	24.4417
[37, Algorithm 3.1]	25.9610	0.7169	22.6684
[15, Algorithm 1]	24.6059	0.7412	19.0017
Algorithm 5.1	26.6238	0.7551	20.4592

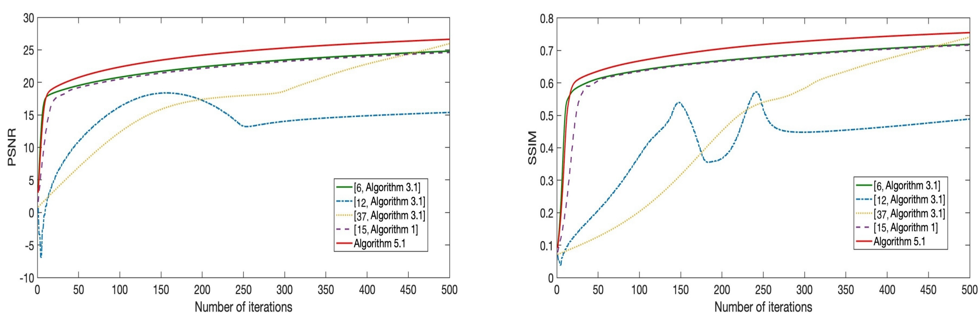
From Table 2, it appears that our Algorithm 5.1 has more efficient than others since PSNR and SSIM values of Algorithm 5.1 have the highest number in the experiment for 500 iterations.

We next demonstrate the figures of restored images.



(A) PSNR=24.8103 (B) PSNR=18.3787 (C) PSNR=25.9610 (D) PSNR=24.6059 (E) PSNR=26.6238

FIGURE 3. (A),(B),(C),(D) and (E) are the restored images by [6, Algorithm 3.1], [12, Algorithm 3.1], [37, Algorithm 3.1], [15, Algorithm 1] and Algorithm 5.1, respectively



(A) PSNR plotting (B) SSIM plotting

FIGURE 4. Graphs of PSNR and SSIM for each algorithm

In Figures 3 and 4, it is shown that Algorithm 5.1 outperforms other algorithms in terms of PSNR and SSIM.

5.3. **Application to the diabetes prediction.** In this section, we apply Algorithm 5.1 for data classification by using the extreme learning machine (ELM).

Consider a training set $\{(x_n, y_n) : x_n \in \mathbb{R}^N, y_n \in \mathbb{R}^M, n = 1, 2, 3, ..., W\}$ where W is the distinct samples, x_n represents an input training data and y_n is a training target. In the context of ELM with a single hidden layer, the output at the i-th hidden node is defined as:

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

where U is an activation function, a_i and b_i are the weight and the bias at the i-th hidden node, respectively.

The output function of 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 the optimal output weight at the i-th hidden node. The hidden layer output matrix A is defined as follows:

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

The main aim of ELM is to compute an optimal weight $\omega = [\omega_1, ..., \omega_L]^T$ such that $A\omega = \chi$, where $\chi = [t_1, ..., t_W]^T$ is the training target data. A model used to find the solution ω can be transformed into a constraint minimization problem as follows:

$$\min_{\omega \in C} \frac{1}{2}\|A\omega - \chi\|^2, \ C = \{\omega | \|\omega\|_1 \leq \xi\}, \tag{5.21}$$

where ξ is a positive constant. In particular, if $C = \{\omega | \|\omega\|_1 \leq \xi\}$ and $Q = \{\chi\}$ then (5.21) can be considered as the SFP.

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

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

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 are defined by J.

The precision and recall can justify performance evaluation in classification. Recall (Rec) also known as the True Rate, measures the accuracy of predictions in positive classes and represents the percentage of correctly predicted positive observations. Accuracy (Acc), prediction (Pre) and F1-score can be calculated using the equation below: [18]

Pre = $\dfrac{\text{TP}}{\text{TP} + \text{FP}} \times 100\%$

Rec = $\dfrac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%$

Acc = $\dfrac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \times 100\%$

F1-score = $\dfrac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$,

where a confusion matrix for original and predicted classes are shown in terms of TP = True Positive, TN =True Negative, FP = False Positive and FN= False negative.

Firstly, we mention about diabetes which is a chronic, metabolic condition characterized by elevated levels of blood glucose (or blood sugar). Over time, this condition can severely damage vital organ such as the heart, blood vessels, eyes, kidneys and nerves. The most prevalent form is type 2 diabetes, typically diagnosed in adults, which occurs when the body either becomes resistant to insulin or doesn't produce enough. Over the past three decades the prevalence of type 2 diabetes has risen dramatically in countries of all income levels. Early predict of diabetes is essential because it can prevent severe damage to many of the body's systems.

In this numerical experiments, we use the PIMA Indians diabetes dataset [39]. This dataset comprises 768 pregnant female patients which 500 were non-diabetics and 268 were diabetics. This dataset contains 9 attributes are Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, Age and Outcome (the predicted attribute). We show visualization of PIMA Indians diabetes dataset in Table 3.

TABLE 3. Overview of all attributes used in training the models

Attributes	Mean	SD	CV	Min	Max
The number of pregnancies	3.85	3.37	11.35	0	17
Plasma glucose	120.89	31.97	1022.25	0	199
Diastolic blood pressure (mmHg)	69.11	19.36	374.65	0	122
Triceps skin fold thickness (mm)	20.54	15.95	254.47	0	99
2-Hour serum insulin (Mu.U/ml)	79.80	115.24	13281.18	0	846
BMI	31.99	7.88	62.16	0	67.1
Diabetes pedigree function	0.47	0.33	0.11	0.08	2.42
Age	33.24	11.76	138.30	21	81

SD: Standard deviation; CV: Coefficient of variation.

In particular, we apply our algorithms to optimize weight parameter in training data for machine learning by using 5-fold cross-validation in extreme learning machine (ELM). We start computation by setting the activation function as sigmoid, hidden nodes $L = 60$, and the parameter is $\xi = 0.7$. We compare our algorithm with the results of Byrne et al. [6] and Chuang [12].

For comparison, we set $x_{-1} = x_0 = x_1 = (1, 1, 1, ..., 1)$, $\gamma = 1$, $\ell = 0.3$, $\mu = 0.7$, $\alpha_n = \frac{1}{(n+1)^7}$ and $\delta_n = \frac{1}{(n+1)^6}$ in Algorithm 5.1.

In [6, Algorithm 3.1], we set $x_1 = (1, 1, 1, ..., 1)$ and $\lambda = \frac{0.1}{\|A\|^2}$.

In [12, Algorithm 3.1], we set $x_0 = (1, 1, 1, ..., 1)$, $x_1 = (0, 0, 0, ..., 0)$, $\theta = 0.5$, $\lambda_n = \frac{\rho}{\|A\|^2}$ and $\rho = 0.1$, $\theta_n = \begin{cases} \min\{\frac{1}{n^3\|x_n - x_{n-1}\|^2}, \theta\} & \text{if } x_n \neq x_{n-1}, \\ \theta & \text{otherwise.} \end{cases}$

We compare the performance of each algorithm for 100 and 200 iterations. The numerical results are reported in Table 4 and Table 5, respectively.

TABLE 4. The performance of each algorithm for 100 iterations

Algorithms	Precision	Recall	F1-score	Accuracy
[6, Algorithm 3.1]	78.56	100	87.99	78.56
[12, Algorithm 3.1]	78.56	100	87.99	78.56
Algorithm 5.1	79.34	100	88.48	79.54

TABLE 5. The performance of each algorithm for 200 iterations

Algorithms	Precision	Recall	F1-score	Accuracy
[6, Algorithm 3.1]	78.56	100	87.99	78.56
[12, Algorithm 3.1]	79.08	100	88.32	79.21
Algorithm 5.1	79.34	100	88.48	79.54

In Tables 4 and 5, we show that our algorithm obtain the best accuracy at 100 and 200 iterations. We observe that Algorithm 5.1 has a better accuracy than other algorithms.

Next, we show the performance with the highest accuracy of each algorithm for prediction in terms of iterations. The comparison are presented in Table 6. From Table 6, we observe that Algorithm 5.1 has less iterations than [6, Algorithm 3.1] and [12, Algorithm 3.1] with the same precision, recall, F1-score and accuracy. This shows that our algorithm has the highest probability of classification for the PIMA Indians diabetes dataset compared to other algorithms.

TABLE 6. The performance of each algorithm

Algorithms	Iter.	Trainning time	Pre	Rec	F1-score	Acc
[6, Algorithm 3.1]	457	0.4304	79.34	100	88.48	79.54
[12, Algorithm 3.1]	257	0.4791	79.34	100	88.48	79.54
Algorithm 5.1	54	0.5231	79.34	100	88.48	79.54

Next, we present graphs of accuracy and loss of training data and testing data for overfitting of Algorithm 5.1.

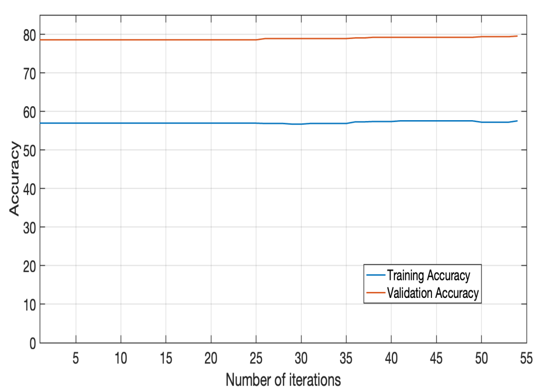


FIGURE 5. Accuracy of Algorithm 5.1

Figures 5 and 6 show the convergence behaviour of accuracy and loss of Algorithm 5.1. We see that it has a high gap between training and validation. This shows that a few training data set are inadequated to train the model. However, graphs of accuracy and loss tend in the same way, which show that Algorithm 5.1 can still classify the PIMA Indians diabetes dataset.

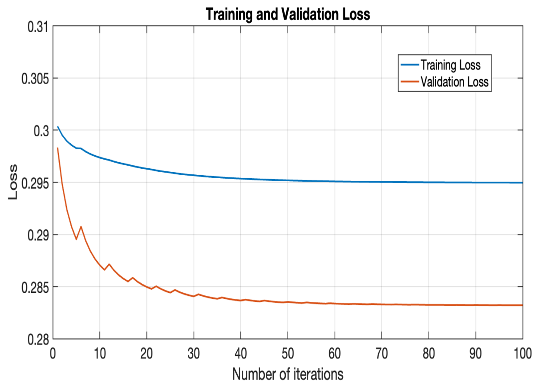


FIGURE 6. Loss of Algorithm 5.1

6. CONCLUSIONS

In this paper, we have proposed iterative schemes with new inertial technique and linesearch stepsize for split variational inclusion problem. Under some conditions, the weak convergence theorem are obtained in the framework of Hilbert spaces. Applications of our obtained results to split feasibility problem have been provided. Furthermore, we apply the proposed algorithm to image deblurring and the data classification of diabetes prediction.

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] Alvarez, F. Weak convergence of a relaxed and inertial hybrid projection-proximal point algorithm for maximal monotone operators in Hilbert space. *SIAM J. Optim.* **14** (2004), no. 3, 773–782.
[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), 3–11.
[3] 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.
[4] Bauschke, H. H.; Combettes, P. L. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*; Springer: New York, NY, USA; Volume 408, 2011.
[5] Beck, A.; Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.* **2** (2009), no. 1, 183–202.
[6] Byrne, C.; Censor, Y.; Gibali, A.; Reich, S. The split common null point problem. *J. Nonlinear Convex Anal.* **13** (2012), no. 4, 759–775.
[7] Censor, Y.; Gibali, A.; Reich, S. Algorithms for the split variational inequality problem. *Numer. Algor.* **59** (2012), 301–323.
[8] Censor, Y.; Elfving, T. A multiprojection algorithm using Bregman projections in a product space. *Numer. Algor.* **8** (1994), no. 2, 221–239.
[9] Chambolle, A.; Lions, P. L. Image recovery via total variation minimization and related problems. *Numer. Math. (Heidelb)* **76** (1997), 167–188.
[10] Cholamjiak, W.; Suparatulatorn, R. Strong convergence of a modified extragradient algorithm to solve pseudomonotone equilibrium and application to classification of diabetes mellitus. *Chaos Solitons Fractals* **168** (2023), 113108.

[11] Chuang, C. S. Strong convergence theorems for the split variational inclusion problem in Hilbert spaces. *J. Fixed Point Theory Appl.* (2013), 1–20.

[12] Chuang, C. S. Hybrid inertial proximal algorithm for the split variational inclusion problem in Hilbert spaces with applications. *Optimization* **66** (2017), no. 5, 777–792.

[13] Dang, Y.; Gao, Y. The strong convergence of a KM-CQ-like algorithm for a split feasibility problem. *Inverse Probl.* **27** (2010), no. 1, 015007.

[14] Dong, Q. L.; Cho, Y. J.; Rassias, T. M. General inertial Mann algorithms and their convergence analysis for nonexpansive mappings. *Appl. Non. Analy.* (2018), 175–191.

[15] Dong, Q. L.; Liu, L.; Qin, X.; Yao, J. C. An alternated inertial general splitting method with linearization for the split feasibility problem. *Optimization* **72** (2023), no. 10, 2585–2607.

[16] Goebel, K.; Kirk, W. A. *Topics in metric fixed point theory*. Cambridge: Cambridge University Press; 1990. (Cambridge studies in advanced mathematics; vol. 28).

[17] Gabeleh, M.; Shahzad, N. Existence and uniqueness of a solution for some nonlinear programming problems. *Mediterr. J. Math.* **12** (2015), 133–146.

[18] Han, J.; Kamber, M.; Pei, J. *Data mining: concepts and techniques*. Waltham, MA. Morgan Kaufman Publishers, **10** (2012), 978–1.

[19] Hanjing, A.; Suantai, S. A fast image restoration algorithm based on a fixed point and optimization method. *Mathematics* **8** (2020), no. 3, 378.

[20] Kazmi, K. R.; Rizvi, S. H. An iterative method for split variational inclusion problem and fixed point problem for a nonexpansive mapping. *Optim. Lett.* **8** (2014), 1113–1124.

[21] Kesornprom, S.; Cholamjiak, P. Proximal type algorithms involving linesearch and inertial technique for split variational inclusion problem in hilbert spaces with applications. *Optimization* **68** (2019), no. 12, 2369–2395.

[22] Martinet, B. Bréve communication. Régularisation d'inéquations variationnelles par approximations successives. *Revue française d'informatique et de recherche opérationnelle. Série rouge* **4(R3)** (1970), 154–158.

[23] Majee, P.; Nahak, C. On inertial proximal algorithm for split variational inclusion problems. *Optimization* **67** (2018), no. 10, 1701–1716.

[24] Marino, G.; Xu, H.K. Convergence of generalized proximal point algorithms. *Commun. Pure Appl. Anal.* **3** (2004), no. 4, 791–808.

[25] Moudafi, A. Split monotone variational inclusions. *J. Optim. Theory Appl.* **150** (2011), 275–283.

[26] Nguyen, T. L.; Shin, Y. *Deterministic sensing matrices in compressive sensing: A survey*. The Scientific World Journal, 2013.

[27] Ogbuisi, F. U.; Shehu, Y.; Yao, J. C. Relaxed Single Projection Methods for Solving Bilevel Variational Inequality Problems in Hilbert Spaces. *Netw. Spat. Econ.* (2023), 1–38.

[28] Opial, Z. Weak convergence of the sequence of successive approximations for nonexpansive mappings. *Bull. Am Math Soc.* **73** (1967), no. 4, 591–597.

[29] 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.

[30] Polyak, B. T. Some methods of speeding up the convergence of iteration methods. *Ussr Comput. Math. Math. Phy.* **4** (1964), no. 5, 1–17.

[31] Qin, X.; Yao, J. C. A viscosity iterative method for a split feasibility problem. *J. Nonlinear Convex Anal.* **20** (2019), no. 8, 1497–1506.

[32] Rockafellar, R. T. Monotone operators and the proximal point algorithm. *SIAM J. Control Optim.* **14** (1976), no. 5, 877–898.

[33] Tan, B.; Qin, X.; Yao, J. C. Strong convergence of self-adaptive inertial algorithms for solving split variational inclusion problems with applications. *J. Sci. Comput.* **87** (2021), no. 1, 20.

[34] Thong, D. V.; Dung, V. T.; Cho, Y. J. A new strong convergence for solving split variational inclusion problems. *Numer. Algor.* **86** (2021), 565–591.

[35] Thung, K. H.; Raveendran, P. A survey of image quality measures. *In 2009 international conference for technical postgraduates (TECHPOS)* (2009, December), (pp. 1-4). IEEE.

[36] Van Long, L.; Viet Thong, D.; Tien Dung, V. New algorithms for the split variational inclusion problems and application to split feasibility problems. *Optimization* **68** (2019), no. 12, 2339–2367.

[37] Vinh, N. T.; Hoai, P. T.; Dung, L. A.; Cho, Y. J. A New Inertial Self-adaptive Gradient Algorithm for the Split Feasibility Problem and an Application to the Sparse Recovery Problem. *Acta Mathematica Sinica, English Series* **39** (2023), no. 12, 2489–2506.

[38] Wang, Z.; Bovik, A. C.; Sheikh, H. R.; Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process* **13** (2004), no. 4, 600–612.

[39] Aman, C. Predict Diabetes. *https://www.kaggle.com/datasets/whenamancodes/predict-diabities*, 2022.

[1]Research Center in Optimization and Computational Intelligence for Big Data Prediction
Department of Mathematics, Faculty of Science
Chiang Mai University, Chiang Mai, 50200, Thailand
Email address: suthep.s@cmu.ac.th

[2]Office of Research Administration, Chiang Mai University
Chiang Mai 50200, Thailand
Email address: suparat.ke@gmail.com

[3]School of Science, University of Phayao
Phayao 56000, Thailand
Email address: prasit.ch@up.ac.th
Email address: papatsara.inkrong@gmail.com