J. Nonlinear Funct. Anal. 2025 (2025) 22 https://doi.org/10.23952/jnfa.2025.22

MATHRES Journal of Nonlinear Functional Analysis
Available online at http://jnfa.mathres.org

A SPLITTING ALGORITHM WITH INERTIAL AND CORRECTION TERMS FOR SOLVING SPLIT VARIATIONAL INCLUSION PROBLEMS AND ITS APPLICATIONS

SUPARAT KESORNPROM[1,2], SUTHEP SUANTAI[2,3], PRASIT CHOLAMJIAK[4,*]

[1]Office of Research Administration, Chiang Mai University, Chiang Mai 50200, Thailand
[2]Department of Mathematics, Faculty of Science, Chiang Mai University, Chiang Mai 50200, Thailand
[3]Excellence Center in Optimization and Computational Intelligence for Big Data Prediction, Department of Mathematics, Faculty of Science, Chiang Mai University, Chiang Mai 50200, Thailand
[4]School of Science, University of Phayao, Phayao 56000, Thailand

Abstract. In this paper, we present a novel splitting algorithm to solve a split variational inclusion problem by using inertial and correction terms, alongside a self-adaptive stepsize technique. The addition of inertial and correction terms significantly accelerates convergence while ensuring weak convergence under some conditions. Numerical experiments demonstrate that our method outperforms existing algorithms in terms of performance and efficiency. Furthermore, we apply our algorithm to solve split feasibility problems and give applications to image recovery and data classification.

Keywords. Correction term; Data classification; Self adaptive method; Splitting algorithm; Variational inclusion problem.

2020 MSC. 46N10, 49J40, 65K10.

1. INTRODUCTION

The study of variational inclusion problems has become a central theme in optimization, equilibrium theory, and mathematical programming due to its applicability in modeling complex systems. Fundamentally, this framework generalizes several mathematical problems, including saddle point problems, variational inequalities, complementarity problems, and fixed-point problems, making it a powerful tool for addressing diverse practical applications; see, e.g., [1, 4, 5, 7, 14, 17, 28, 29]. In practical terms, split variational inequality problems (SVIP) are beneficial in are two Hilbert spaces, H_1 and H_2, with the aid of a bounded and linear operator A. For example, SVIP are relevant in fields where solutions must align with different functional

*Corresponding author.
E-mail address: suparat.ke@gmail.com (S. Kesornprom), suthep.s@cmu.ac.th (S. Suantai), prasit.ch@up.ac.th (P. Cholamjiak).

Received February 13, 2025; Accepted June 21, 2025.

constraints across multiple domains, such as signal processing, image reconstruction, and data science; see, e.g., [2, 9, 13, 21, 26, 27].

The concept of the SVIP was first defined by Censor et al. [7] expanding split feasibility problems (SFP) into variational inequality problems. The framework presents a variational inclusion on two Hilbert spaces: given a convex and closed subset $C \subset H_1$, a mapping $f : H_1 \to H_1$, and a linear operator $A : H_1 \to H_2$, the objective is to identify a point $x^* \in C$ such that

$$\langle f(x^*), x - x^* \rangle \geq 0, \quad \forall x \in C$$

and simultaneously, that the transformed point $y^* = Ax^* \in H_2$ satisfies

$$\langle g(y^*), y - y^* \rangle \geq 0, \quad \forall y \in Q,$$

where $Q \subset H_2$ and $g : H_2 \to H_2$ are similarly defined. This formulation enables coupling of two variational inequality problems via A, the associated operator. Censor et al. [7] employed projection and resolvent operators within an averaged operator framework and the Krasnosel'skii-Mann theorem to guarantee weak convergence of their algorithms under specific conditions.

In 2011, Moudafi [19] introduced the split monotone variational inclusion problem (SMVI), which generalizes the SVIP by using maximal monotone mappings $B_1 : H_1 \to 2^{H_1}$ and $B_2 : H_2 \to 2^{H_2}$. The problem is now to find a solution $x^* \in H_1$ such that

$$0 \in f(x^*) + B_1(x^*) \quad \text{and} \quad 0 \in g(Ax^*) + B_2(Ax^*).$$

In 2011, Byrne et al. [4] developed an iterative scheme for solving the SVIP in Hilbert spaces. Their iterative was defined as follows:

$$x_{n+1} = J_\beta^{B_1}(x_n - \gamma A^*(I - J_\beta^{B_2})Ax_n),$$

where $J_\beta^B(x) = (I + \beta B)^{-1}(x)$ for each $x \in H_1$ and β and γ are real positive numbers.

In 2019, Kesornprom and Cholamjiak [16] considered an inertial scheme for the SVIP framework, inspired by acceleration methods from dynamical systems and Nesterov's optimization method. Their inertial technique leads to the following iterative form:

$$x_{n+1} = J_\beta^{B_1}(x_n + \theta_n(x_n - x_{n-1}) - \gamma A^*(I - J_\beta^{B_2})Ax_n),$$

where $\theta_n \in [0, 1)$ is the inertial parameter that controls the contribution of past iterations. Kesornprom and Cholamjiak [16] demonstrated that, under suitable monotonicity conditions, the inertial scheme can achieve weak convergence.

Another recent development is the self-adaptive stepsize, which introduced by López et al. [18]. The self-adaptive stepsize reads as follows: $\gamma_n = \frac{\rho_n f(x_n)}{\|\nabla f(x_n)\|^2}$, where $\rho_n \in (0, 4)$, $f(x) = \frac{1}{2}\|(I - P_Q)Ax\|^2$, $\nabla f(x) = A^*(I - P_Q)Ax$, and P_C and P_Q are the orthogonal projections on C and Q, the convex and closed sets, respectively. This stepsize enables the scheme to ensure convergence even when operator norms are not known.

In 2024, Izuchukwu et al. [15] proposed a proximal point algorithm with inertial terms and two correction terms as follows: For $w_{-1} = w_{-2}, x_0, x_{-1} \in H$, set $n = 0$ and then compute

$$\begin{cases} w_n = x_n + \alpha(x_n - x_{n-1}) + \delta(1 + \alpha)(w_{n-1} - x_n) - \alpha\delta(w_{n-2} - x_{n-1}), \\ x_{n+1} = J_\beta^B(w_n), \end{cases}$$

where $0 \leq \alpha < \frac{\varepsilon}{1+\varepsilon}$, $\frac{(1+\varepsilon)\alpha}{1+\alpha\varepsilon} < \delta < 1$ for some $\varepsilon > 0$. The inertial term $\alpha(x_n - x_{n-1})$ and two correction terms $\delta(1+\alpha)(w_{n-1} - x_n)$ and $\alpha\delta(w_{n-2} - x_{n-1})$ are added to the iterative scheme to improve on the convergence speed of algorithm.

In order to further improve on the convergence speed of the splitting algorithm with inertial and correction terms for solving SVIP with the following contributions:

- We introduce a new splitting method with corrected and inertial terms and self-adaptive stepsizes for solving the SVIP. Weak convergence of the generated sequence are given under assumptions.

- We present numerical experiments to support the efficiency of our new algorithm.

- We give applications of our results to split feasibility problems arising from image deblurring and date classification.

2. Preliminaries and lemmas

In this section, we provide some basic definitions and lemmas which are used in the sequel. Let H be a real Hilbert space. We use the symbol $\rightharpoonup$ stands for the weak convergence. Recall that a mapping $T : H \to H$ is said to be

(1) nonexpansive if, for all $x, y \in H$, $\|Tx - Ty\| \leq \|x - y\|$;

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

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 $\|Tx - Ty\| \leq L\|x - y\|$ for all $x, y \in H$.

A set-valued mapping $B : H \to 2^H$ is called monotone if, for all $x, y \in H$

$$\langle u - v, x - y\rangle \geq 0, \ \forall u \in Bx \text{ and } v \in By.$$

For a set-valued mapping B, graph(B) is defined as graph$(B) := \{(x,u) \in H \times H : u \in B(x)\}$. A monotone mapping $B : H \to 2^H$ is said to be maximal if the graph(B) is not properly contained in the graph of any other monotone mapping. Let $B : H \to 2^H$ be a set-valued maximal monotone operator. The resolvent operator $J_\beta^B : H \to H$ associated with B is defined by

$$J_\beta^B(x) = (I + \beta B)^{-1}(x), \ \forall x \in H,$$

where $\beta > 0$. It is well known that the resolvent operator is single-valued and firmly nonexpansive.

In order to study the SVIP, we recall some lemmas which are needed in our proof. We denote by $B^{-1}(0) = \{x \in H : 0 \in Bx\}$, $\mathscr{D}(T)$ the domain of T and $Fix(T)$ the fixed point set of T, that is, $Fix(T) = \{x \in H : x = Tx\}$.

Lemma 2.1. *(Demiclosedness Principle [12]) Let C be a convex, closed, and nonempty subset of a real Hilbert space H and let $T : C \to C$ be a nonexpansive mapping. If $x_n \rightharpoonup x \in C$ and $\lim_{n\to\infty} \|x_n - Tx_n\| = 0$, then $x = Tx$.*

Lemma 2.2. [3] *Let H be a real Hilbert space and $B : H \to 2^H$ be a set-valued maximal monotone mapping for each $x, y \in H$, each $z \in B^{-1}(0)$, and each $\beta > 0$. Then*

(i) $\langle x - J_\beta^B x, z - J_\beta^B x\rangle \leq 0$;

(ii) $\|J_\beta^B x - J_\beta^B y\|^2 \leq \langle J_\beta^B x - J_\beta^B y, x - y\rangle$;

(iii) $\|J_\beta^B x - z\|^2 \leq \|x - z\|^2 - \|J_\beta^B x - x\|^2$.

Lemma 2.3. [8, 20] *Let H be a real Hilbert space and $B : H \to 2^H$ be a set-valued maximal monotone mapping. Then*

(i) J_β^B is a single-valued and firmly nonexpansive mapping for each $\beta > 0$;

(ii) $\mathscr{D}(J_\beta^B) = H$ and $Fix(J_\beta^B) = \{x \in \mathscr{D}(B) : 0 \in Bx\}$;

(iii) $\|x - J_\beta^B x\| \leq \|x - J_\gamma^B x\|$ for all $0 < \beta \leq \gamma$ and for all $x \in H$;

(iv) If $B^{-1}(0) \neq \emptyset$, then $\|x - J_\beta^B x\|^2 + \|J_\beta^B - x^*\|^2 \leq \|x - x^*\|^2$ for each $x \in H$, $x^* \in B^{-1}(0)$, and $\beta > 0$.

(v) If $B^{-1}(0) \neq \emptyset$, then $\langle x - J_\beta^B x, J_\beta^B x - w\rangle \geq 0$ for each $x \in H$, $w \in B^{-1}(0)$, and $\beta > 0$.

The next lemma gives a crucial characterization of the solution sets of the SVIP and the fixed point sets of the resolvent operator.

Lemma 2.4. [8] *Let H_1 and H_2 be real Hilbert spaces and $A : H_1 \to H_2$ be a bounded and linear operator. Let $\beta > 0$, $\gamma > 0$, $B_1 : H_1 \to 2^{H_1}$, and $B_2 : H_2 \to 2^{H_2}$ be set-valued maximal monotone mappings. For any $x^* \in H_1$,*

(i) if x^* is a solution to (SVIP), then $J_\beta^{B_1}(x^* - \gamma A^*(I - J_\beta^{B_2})Ax^*) = x^*$;

(ii) if $J_\beta^{B_1}(x^* - \gamma A^*(I - J_\beta^{B_2})Ax^*) = x^*$ and the solution set of (SVIP) is nonempty, then x^* is a solution to (SVIP).

Lemma 2.5. [8] *Let H_1 and H_2 be real Hilbert spaces, $A : H_1 \to H_2$ be a bounded linear operator, and $\beta > 0$. Let $B : H_2 \to 2^{H_2}$ be a set-valued maximal monotone mapping. Define a mapping $T : H_1 \to H_1$ by $Tx := A^*(I - J_\beta^B)Ax$ for each $x \in H_1$. Then*

(i) $\|(I - J_\beta^B)Ax - (I - J_\beta^B)Ay\|^2 \leq \langle Tx - Ty, x - y\rangle$ for all $x, y \in H_1$;

(ii) $\|A^*(I - J_\beta^B)Ax - A^*(I - J_\beta^B)Ay\|^2 \leq \|A\|^2 \cdot \langle Tx - Ty, x - y\rangle$ for all $x, y \in H_1$.

We also need the following tools in convergence analysis.

Lemma 2.6. [23] *Let $a_{n+1} \leq (1 + c_n)a_n + b_n$, where $\{a_n\}$, $\{b_n\}$, and $\{c_n\}$ are real positive sequences. If $\Sigma_{n=1}^{\infty} c_n < +\infty$ and $\Sigma_{n=1}^{\infty} b_n < +\infty$, then $\lim_{n\to+\infty} a_n$ exists.*

Lemma 2.7. *(Opial theorem [22]) Let C be a nonempty subset of a real Hilbert space H and $\{x_n\}$ be a sequence in H that satisfies the following properties:*

(i) $\lim_{n\to\infty} \|x_n - x\|$ exists for each $x \in C$;

(ii) every sequential weak limit point of $\{x_n\}$ is in C.

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

3. Main Results

In this section, we present a splitting algorithm with inertial and correction terms for solving the split variational inclusion problem. Let H_1 and H_2 be real Hilbert spaces. We define $A : H_1 \to H_2$ as a linear and bounded operator and denote A^* by the adjoint operator of A. Let $B_1 : H_1 \to 2^{H_1}$ and $B_2 : H_2 \to 2^{H_2}$ be maximal monotone operators. We denote the solution set of SVIP by Ω, where Ω is nonempty. We assume that

(I) $0 < \liminf \rho_n \leq \limsup \rho_n < 2$;

(II) Either $\alpha_n = 0$, $\forall k \geq 0$ or $-1 < d_1 \leq \alpha_n \leq \alpha_{n+1} \leq d_2 < 0$.

Algorithm 3.1. *Let $\rho_n \in (0,2)$, $\alpha_n \in (-1,0]$, $\delta \in [0,1)$ and $\{\beta_n\}$ be a sequence in $(0,\infty)$. Take arbitrarily $w_{-1}, w_{-2}, x_{-1}, x_0 \in H_1$ and $\lambda_n, \gamma_n > 0$. Set $n = 0$ and define*

$$\begin{aligned} w_n &= x_n + \alpha_n(x_n - x_{n-1}) + \delta(1+\alpha_n)(w_{n-1} - x_n) - \alpha_n\delta(w_{n-2} - x_{n-1}) \\ y_n &= w_n - \gamma_n A^*(I - J_{\beta_n}^{B_2})Aw_n) \\ x_{n+1} &= J_{\beta_n}^{B_1}(y_n - \lambda_n A^*(I - J_{\beta_n}^{B_2})Ay_n) \end{aligned} \tag{3.1}$$

where

$$\lambda_n = \frac{\rho_n \|(I - J_{\beta_n}^{B_2})Ay_n\|^2}{\|A^*(I - J_{\beta_n}^{B_2})Ay_n\|^2} \text{ and } \gamma_n = \frac{\rho_n \|(I - J_{\beta_n}^{B_2})Aw_n\|^2}{\|A^*(I - J_{\beta_n}^{B_2})Aw_n\|^2}.$$

Lemma 3.2. *The sequences $\{w_n\}$ and $\{x_n\}$ generated by Algorithm 3.1 are bounded.*

Proof. Let $x^* \in \Omega$. Then $x^* \in B_1^{-1}(0)$ and $Ax^* \in B_2^{-1}(0)$. Moreover, we have $J_{\beta_n}^{B_2}Ax^* = Ax^*$. Define

$$z_n = x_n + \delta(w_{n-1} - x_n). \tag{3.2}$$

It follows that $x_n = \frac{1}{1-\delta}z_n - \frac{\delta}{1-\delta}w_{n-1}$. Thus

$$\begin{aligned} x_n - z_n &= \frac{1}{1-\delta}[z_n - \delta w_{n-1} - z_n + \delta z_n] \\ &= \frac{\delta}{1-\delta}[(x_n + \delta(w_{n-1} - x_n)) - w_{n-1}] \\ &= \delta(x_n - w_{n-1}). \end{aligned}$$

Using Lemma 2.2(iii), we obtain

$$\begin{aligned} \|x_{n+1} - x^*\|^2 &= \|J_{\beta_n}^{B_1}(y_n - \lambda_n A^*(I - J_{\beta_n}^{B_2})Ay_n) - x^*\|^2 \\ &\leq \|y_n - \lambda_n A^*(I - J_{\beta_n}^{B_2})Ay_n - x^*\|^2 - \|x_{n+1} - y_n + \lambda_n A^*(I - J_{\beta_n}^{B_2})Ay_n\|^2 \\ &\leq \|y_n - x^*\|^2 - 2\lambda_n\langle y_n - x^*, A^*(I - J_{\beta_n}^{B_2})Ay_n\rangle + \lambda_n^2\|A^*(I - J_{\beta_n}^{B_2})Ay_n\|^2 \\ &\quad - \|x_{n+1} - y_n + \lambda_n A^*(I - J_{\beta_n}^{B_2})Ay_n\|^2. \end{aligned} \tag{3.3}$$

Using Lemma 2.5(i), we have

$$\begin{aligned} \langle A^*(I - J_{\beta_n}^{B_2})Ay_n, y_n - x^*\rangle &= \langle (I - J_{\beta_n}^{B_2})Ay_n, Ay_n - Ax^*\rangle \\ &= \langle (I - J_{\beta_n}^{B_2})Ay_n - (I - J_{\beta_n}^{B_2})Ax^*, Ay_n - Ax^*\rangle \\ &\geq \|(I - J_{\beta_n}^{B_2})Ay_n\|^2. \end{aligned} \tag{3.4}$$

Observe that

$$\langle A^*(I - J_{\beta_n}^{B_2})Aw_n, w_n - x^*\rangle \geq \|(I - J_{\beta_n}^{B_2})Aw_n\|^2. \tag{3.5}$$

From (3.3), (3.4), and definition of λ_n, we obtain

$$\begin{aligned}\|x_{n+1}-x^*\|^2 &\leq \|y_n-x^*\|^2-2\lambda_n\|(I-J_{\beta_n}^{B_2})Ay_n\|^2+\lambda_n^2\|A^*(I-J_{\beta_n}^{B_2})Ay_n\|^2\\ &\quad-\|x_{n+1}-y_n+\lambda_nA^*(I-J_{\beta_n}^{B_2})Ay_n\|^2\\ &=\|y_n-x^*\|^2-\rho_n(2-\rho_n)\frac{\|(I-J_{\beta_n}^{B_2})Ay_n\|^4}{\|A^*(I-J_{\beta_n}^{B_2})Ay_n\|^2}\\ &\quad-\|x_{n+1}-y_n+\lambda_nA^*(I-J_{\beta_n}^{B_2})Ay_n\|^2.\end{aligned}\tag{3.6}$$

From (3.2), we have

$$z_{n-1}=x_{n-1}+\delta(w_{n-2}-x_{n-1}).\tag{3.7}$$

From the definition of the sequence $\{w_n\}$, we obtain

$$\begin{aligned}w_n &= x_n+\alpha_n(x_n-x_{n-1})+\delta(1+\alpha_n)(w_{n-1}-x_n)-\alpha_n\delta(w_{n-2}-x_{n-1})\\ &= x_n+\delta(w_{n-1}-x_n)+\alpha_n[(x_n+\delta(w_{n-1}-x_n))-(x_{n-1}+\delta(w_{n-2}-x_{n-1}))].\end{aligned}\tag{3.8}$$

Replacing (3.2) and (3.7) in (3.8), we obtain

$$w_n=z_n+\alpha_n(z_n-z_{n-1}).\tag{3.9}$$

From (3.5), we see that

$$\begin{aligned}\|y_n-x^*\|^2 &= \|w_n-x^*\|^2-2\gamma_n\langle w_n-x^*,A^*(I-J_{\beta_n}^{B_2})Aw_n\rangle+\gamma_n^2\|A^*(I-J_{\beta_n}^{B_2})Aw_n\|^2\\ &\leq \|w_n-x^*\|^2-2\gamma_n\|(I-J_{\beta_n}^{B_2})Aw_n\|^2+\gamma_n^2\|A^*(I-J_{\beta_n}^{B_2})Aw_n\|^2\\ &= \|w_n-x^*\|^2-\frac{2\rho_n\|(I-J_{\beta_n}^{B_2})Aw_n\|^4}{\|A^*(I-J_{\beta_n}^{B_2})Aw_n\|^2}+\frac{\rho_n^2\|(I-J_{\beta_n}^{B_2})Aw_n\|^4}{\|A^*(I-J_{\beta_n}^{B_2})Aw_n\|^2}\\ &= \|w_n-x^*\|^2-\rho_n(2-\rho_n)\frac{\|(I-J_{\beta_n}^{B_2})Aw_n\|^4}{\|A^*(I-J_{\beta_n}^{B_2})Aw_n\|^2}.\end{aligned}\tag{3.10}$$

Combining (3.6) and (3.10), we obtain

$$\begin{aligned}\|x_{n+1}-x^*\|^2 &\leq \|w_n-x^*\|^2-\rho_n(2-\rho_n)\frac{\|(I-J_{\beta_n}^{B_2})Aw_n\|^4}{\|A^*(I-J_{\beta_n}^{B_2})Aw_n\|^2}\\ &\quad-\rho_n(2-\rho_n)\frac{\|(I-J_{\beta_n}^{B_2})Ay_n\|^4}{\|A^*(I-J_{\beta_n}^{B_2})Ay_n\|^2}-\|x_{n+1}-y_n+\lambda_nA^*(I-J_{\beta_n}^{B_2})Ay_n\|^2.\end{aligned}\tag{3.11}$$

We see that $\|z_n-w_{n-1}\|^2=(1-\delta)^2\|(x_n-w_{n-1})\|^2$, which together with 3.2 yields that

$$\begin{aligned}\|x_n-x^*\|^2 &= \|\frac{1}{1-\delta}(z_n-x^*)-\frac{\delta}{1-\delta}(w_{n-1}-x^*)\|^2\\ &= \frac{1}{1-\delta}\|z_n-x^*\|^2-\frac{\delta}{1-\delta}\|w_{n-1}-x^*\|^2+\frac{\delta}{(1-\delta)^2}\|z_n-w_{n-1}\|^2\\ &= \frac{1}{1-\delta}\|z_n-x^*\|^2-\frac{\delta}{1-\delta}\|w_{n-1}-x^*\|^2+\delta\|x_n-w_{n-1}\|^2.\end{aligned}\tag{3.12}$$

Adding $\|x_n - x^*\|^2$ in equation (3.11) and using (3.12), we have

$$\begin{aligned}
&\|x_{n+1}-x^*\|^2 - \|x_n - x^*\|^2 \\
\leq\ & \|w_n - x^*\|^2 - \frac{1}{1-\delta}\|z_n - x^*\|^2 + \frac{\delta}{1-\delta}\|w_{n-1} - x^*\|^2 - \delta\|x_n - w_{n-1}\|^2 \\
&-\rho_n(2-\rho_n)\frac{\|(I-J^{B_2}_{\beta_n})Aw_n\|^4}{\|A^*(I-J^{B_2}_{\beta_n})Aw_n\|^2} - \rho_n(2-\rho_n)\frac{\|(I-J^{B_2}_{\beta_n})Ay_n\|^4}{\|A^*(I-J^{B_2}_{\beta_n})Ay_n\|^2} \\
&-\|x_{n+1} - y_n + \lambda_n A^*(I-J^{B_2}_{\beta_n})Ay_n\|^2. \qquad (3.13)
\end{aligned}$$

From (3.9), we also obtain

$$\|w_n - x^*\|^2 = (1+\alpha_n)\|z_n - x^*\|^2 - \alpha_n\|z_{n-1} - x^*\|^2 + \alpha_n(1+\alpha_n)\|z_n - z_{n-1}\|^2. \qquad (3.14)$$

Replacing (3.14) in (3.13), we obtain

$$\begin{aligned}
&\|x_{n+1}-x^*\|^2 - \|x_n - x^*\|^2 \\
&\leq (1+\alpha_n)\|z_n - x^*\|^2 - \alpha_n\|z_{n-1} - x^*\|^2 + \alpha_n(1+\alpha_n)\|z_n - z_{n-1}\|^2 \\
&\quad - \frac{1}{1-\delta}\|z_n - x^*\|^2 + \frac{\delta}{1-\delta}\|w_{n-1} - x^*\|^2 - \delta\|x_n - w_{n-1}\|^2 \\
&\quad - \rho_n(2-\rho_n)\frac{\|(I-J^{B_2}_{\beta_n})Aw_n\|^4}{\|A^*(I-J^{B_2}_{\beta_n})Aw_n\|^2} - \rho_n(2-\rho_n)\frac{\|(I-J^{B_2}_{\beta_n})Ay_n\|^4}{\|A^*(I-J^{B_2}_{\beta_n})Ay_n\|^2} \\
&\quad - \|x_{n+1} - y_n + \lambda_n A^*(I-J^{B_2}_{\beta_n})Ay_n\|^2 \\
&= (\alpha_n - \frac{\delta}{1-\delta})\|z_n - x^*\|^2 - \alpha_n\|z_{n-1} - x^*\|^2 + \alpha_n(1+\alpha_n)\|z_n - z_{n-1}\|^2 \\
&\quad + \frac{\delta}{1-\delta}\|w_{n-1} - x^*\|^2 - \delta\|x_n - w_{n-1}\|^2 \\
&\quad - \rho_n(2-\rho_n)u_n - \rho_n(2-\rho_n)v_n \\
&\quad - \|x_{n+1} - y_n + \lambda_n A^*(I-J^{B_2}_{\beta_n})Ay_n\|^2,
\end{aligned}$$

where

$$u_n = \frac{\|(I-J^{B_2}_{\beta_n})Aw_n\|^4}{\|A^*(I-J^{B_2}_{\beta_n})Aw_n\|^2} \quad \text{and} \quad v_n = \frac{\|(I-J^{B_2}_{\beta_n})Ay_n\|^4}{\|A^*(I-J^{B_2}_{\beta_n})Ay_n\|^2}.$$

Thus

$$\begin{aligned}
&\|x_{n+1}-x^*\|^2 - \|x_n - x^*\|^2 \\
&\leq (\alpha_n - \frac{\delta}{1-\delta})\|z_n - x^*\|^2 - \alpha_n\|z_{n-1} - x^*\|^2 + \alpha_n(1+\alpha_n)\|z_n - z_{n-1}\|^2 \\
&+ \frac{\delta}{1-\delta}[(1+\alpha_{n-1})\|z_{n-1} - x^*\|^2 - \alpha_{n-1}\|z_{n-2} - x^*\|^2 \qquad (3.15) \\
&\quad + \alpha_{n-1}(1+\alpha_{n-1})\|z_{n-1} - z_{n-2}\|^2] - \delta\|x_n - w_{n-1}\|^2 - \rho_n(2-\rho_n)u_n \\
&\quad - \rho_n(2-\rho_n)v_n - \|x_{n+1} - y_n + \lambda_n A^*(I-J^{B_2}_{\beta_n})Ay_n\|^2,
\end{aligned}$$

which implies that

$$\begin{aligned}
&\|x_{n+1}-x^*\|^2-(\alpha_n-\frac{\delta}{1-\delta})\|z_n-x^*\|^2-\alpha_n(1+\alpha_n)\|z_n-z_{n-1}\|^2\\
&\leq \|x_n-x^*\|^2+(\frac{\delta(1+\alpha_{n-1})}{1-\delta}-\alpha_n)\|z_{n-1}-x^*\|^2\\
&\quad+\frac{\delta\alpha_{n-1}(1+\alpha_{n-1})}{1-\delta}\|z_{n-1}-z_{n-2}\|^2-\frac{\delta\alpha_{n-1}}{1-\delta}\|z_{n-2}-x^*\|^2\\
&\quad-\delta\|x_n-w_{n-1}\|^2-\rho_n(2-\rho_n)u_n-\rho_n(2-\rho_n)v_n\\
&\quad-\|x_{n+1}-y_n+\lambda_nA^*(I-J^{B_2}_{\beta_n})Ay_n\|^2\\
&=\|x_n-x^*\|^2+(\frac{\delta}{1-\delta}+\frac{\delta(1+\alpha_{n-1})}{1-\delta}-\alpha_n)\|z_{n-1}-x^*\|^2\\
&\quad+\frac{\delta\alpha_{n-1}(1+\alpha_{n-1})}{1-\delta}\|z_{n-1}-z_{n-2}\|^2-\frac{\delta\alpha_{n-1}}{1-\delta}\|z_{n-2}-x^*\|^2\\
&\quad-\delta\|x_n-w_{n-1}\|^2-\rho_n(2-\rho_n)u_n-\rho_n(2-\rho_n)v_n\\
&\quad-\|x_{n+1}-y_n+\lambda_nA^*(I-J^{B_2}_{\beta_n})Ay_n\|^2.
\end{aligned}$$

It follows that

$$\begin{aligned}
&\|x_{n+1}-x^*\|^2-(\alpha_n-\frac{\delta}{1-\delta})\|z_n-x^*\|^2-\frac{\delta\alpha_{n-1}}{1-\delta}\|z_{n-1}-x^*\|^2-\alpha_n(1+\alpha_n)\|z_n-z_{n-1}\|^2\\
&\leq \|x_n-x^*\|^2-(\alpha_n-\frac{\delta}{1-\delta})\|z_{n-1}-x^*\|^2-\frac{\delta\alpha_{n-1}}{1-\delta}\|z_{n-2}-x^*\|^2-\delta\|x_n-w_{n-1}\|^2\\
&\quad-\rho_n(2-\rho_n)u_n-\rho_n(2-\rho_n)v_n+\frac{\delta\alpha_{n-1}(1+\alpha_{n-1})}{1-\delta}\|z_{n-1}-z_{n-2}\|^2\\
&\quad-\|x_{n+1}-y_n+\lambda_nA^*(I-J^{B_2}_{\beta_n})Ay_n\|^2.
\end{aligned}$$

Since $\alpha_{n-1}\leq\alpha_n$, we obtain

$$\begin{aligned}
&\|x_{n+1}-x^*\|^2-(\alpha_n-\frac{\delta}{1-\delta})\|z_n-x^*\|^2-\frac{\delta\alpha_n}{1-\delta}\|z_{n-1}-x^*\|^2-\alpha_n(1+\alpha_n)\|z_n-z_{n-1}\|^2\\
&\leq \|x_n-x^*\|^2-(\alpha_{n-1}-\frac{\delta}{1-\delta})\|z_{n-1}-x^*\|^2-\frac{\delta\alpha_{n-1}}{1-\delta}\|z_{n-2}-x^*\|^2-\delta\|x_n-w_{n-1}\|^2\\
&\quad-\rho_n(2-\rho_n)u_n-\rho_n(2-\rho_n)v_n+\frac{\delta\alpha_{n-1}(1+\alpha_{n-1})}{1-\delta}\|z_{n-1}-z_{n-2}\|^2\\
&\quad-\|x_{n+1}-y_n+\lambda_nA^*(I-J^{B_2}_{\beta_n})Ay_n\|^2\\
&=\|x_n-x^*\|^2-(\alpha_{n-1}-\frac{\delta}{1-\delta})\|z_{n-1}-x^*\|^2-\frac{\delta\alpha_{n-1}}{1-\delta}\|z_{n-2}-x^*\|^2\\
&\quad-\alpha_{n-1}(1+\alpha_{n-1})\|z_{n-1}-z_{n-2}\|^2+\frac{\alpha_{n-1}(1+\alpha_{n-1})}{1-\delta}\|z_{n-1}-z_{n-2}\|^2-\delta\|x_n-w_{n-1}\|^2\\
&\quad-\rho_n(2-\rho_n)u_n-\rho_n(2-\rho_n)v_n-\|x_{n+1}-y_n+\lambda_nA^*(I-J^{B_2}_{\beta_n})Ay_n\|^2.
\end{aligned}\tag{3.16}$$

Define

$$a_n = \|x_n - x^*\|^2 - (\alpha_{n-1} - \frac{\delta}{1-\delta})\|z_{n-1} - x^*\|^2 - \frac{\delta\alpha_{n-1}}{1-\delta}\|z_{n-2} - x^*\|^2 - \alpha_{n-1}(1+\alpha_{n-1})\|z_{n-1} - z_{n-2}\|^2.$$

It follows from (3.16) that

$$\begin{aligned} a_{n+1} \leq\ & a_n + \frac{\alpha_{n-1}(1+\alpha_{n-1})}{1-\delta}\|z_{n-1} - z_{n-2}\|^2 - \delta\|x_n - w_{n-1}\|^2 \\ & -\rho_n(2-\rho_n)u_n - \rho_n(2-\rho_n)v_n - \|x_{n+1} - y_n + \lambda_n A^*(I - J_{\beta_n}^{B_2})Ay_n\|^2. \end{aligned} \tag{3.17}$$

Hence $a_n \geq 0$ since $\alpha_n \in (-1,0]$ and $\delta \in [0,1)$. Furthermore, from (3.17), one sees that $\lim_{n\to\infty} a_n$ exists and $\{a_n\}$ is bounded. Hence, $\{x_n\}$ is bounded. □

Lemma 3.3. *Let $\{x_n\}$ be generated by Algorithm 3.1. Then $\lim_{n\to\infty}\|x_n - w_n\| = 0$.*

Proof. From (3.17) and $\rho_n \in (0,2)$, we have

$$\lim_{n\to\infty} \frac{\|(I - J_{\beta_n}^{B_2})Aw_n\|}{\|A^*(I - J_{\beta_n}^{B_2})Aw_n\|} = 0. \tag{3.18}$$

Note that

$$\|A^*(I - J_{\beta_n}^{B_2})Aw_n - A^*(I - J_{\beta_n}^{B_2})Ax^*\| \leq \|A\|^2\|w_n - x^*\|.$$

So $\{\|A^*(I - J_{\beta_n}^{B_2})Aw_n\|\}$ is bounded. From (3.18), we obtain $\lim_{n\to\infty}\|(I - J_{\beta_n}^{B_2})Aw_n\| = 0$, this is,

$$\lim_{n\to\infty}\|Aw_n - J_{\beta_n}^{B_2}Aw_n\| = 0. \tag{3.19}$$

We also have

$$\lim_{n\to\infty}\|Ay_n - J_{\beta_n}^{B_2}Ay_n\| = 0. \tag{3.20}$$

Since $\alpha_n \in (-1,0]$, we consider two cases as follow.

Case I. Set $\alpha_n = 0$ in (3.17). From (3.1), we see that $w_n = x_n + \delta(w_{n-1} - x_n)$. If $\delta \in (0,1)$ in (3.17), then $\lim_{n\to\infty}\|x_n - w_{n-1}\| = 0$. Thus

$$\lim_{n\to\infty}\|w_n - x_n\| = \delta\lim_{n\to\infty}\|x_n - w_{n-1}\| = 0$$

If $\delta = 0$, we find from (3.2) $z_n = x_n$ and hence

$$\lim_{n\to\infty}\|w_n - x_n\| = \lim_{n\to\infty}\|x_n + \delta(w_{n-1} - x_n) - x_n\| = 0.$$

It follows that $\lim_{n\to\infty}\|w_n - x_n\| = 0$.

Case II. Set $-1 < d_1 \leq \alpha_n \leq \alpha_{n+1} \leq d_2 < 0$ in (3.17). If $\delta \in (0,1)$, then $\lim_{n\to\infty}\|z_{n-1} - z_{n-2}\| = 0$ and $\lim_{n\to\infty}\|x_n - w_{n-1}\| = 0$. If $\delta = 0$, then $\lim_{n\to\infty}\|x_n - w_{n-1}\| = 0$. Thus, from $\delta \in [0,1)$, we obtain $\lim_{n\to\infty}\|x_n - w_{n-1}\| = 0$, which together with (3.2) yields

$$\lim_{n\to\infty}\|z_n - x_n\| = \delta\lim_{n\to\infty}\|w_{n-1} - x_n\| = 0.$$

From (3.8), we have

$$\lim_{n\to\infty}\|w_n - z_n\| = \lim_{n\to\infty}|\alpha_n|\|z_n - z_{n-1}\| = 0.$$

Thus

$$\lim_{n\to\infty}\|x_n-w_n\| \le \lim_{n\to\infty}\|x_n-z_n\| + \lim_{n\to\infty}\|z_n-w_n\| = 0. \tag{3.21}$$

Therefore, we can conclude that $\lim_{n\to\infty}\|w_n-x_n\| = 0$. □

Theorem 3.4. *The sequence $\{x_n\}$ generated by Algorithm 3.1 converges weakly to a solution in Ω.*

Proof. From the definition of y_n and (3.19), we have

$$\lim_{n\to\infty}\|y_n-w_n\| = \lim_{n\to\infty}\gamma_n\|A^*(I-J_{\beta_n}^{B_2})Aw_n\| = \lim_{n\to\infty}\frac{\rho_n\|(I-J_{\beta_n}^{B_2})Aw_n\|^2}{\|A^*(I-J_{\beta_n}^{B_2})Aw_n\|} = 0.$$

From (3.19), (3.21), and definition of y_n, we have

$$\begin{aligned}\lim_{n\to\infty}\|y_n-x_n\| &= \lim_{n\to\infty}\|w_n-\gamma_nA^*(I-J_{\beta_n}^{B_2})Aw_n-x_n\|\\ &\le \lim_{n\to\infty}\|w_n-x_n\| + \lim_{n\to\infty}\gamma_n\|A^*(I-J_{\beta_n}^{B_2})Aw_n\|\\ &= 0.\end{aligned} \tag{3.22}$$

From definition of x_{n+1}, (3.20), and (3.22), we have

$$\begin{aligned}\lim_{n\to\infty}\|x_{n+1}-J_{\beta_n}^{B_1}x_n\| &= \|J_{\beta_n}^{B_1}(y_n-\lambda_nA^*(I-J_{\beta_n}^{B_2})Ay_n)-J_{\beta_n}^{B_1}x_n\|\\ &\le \lim_{n\to\infty}\|y_n-\lambda_nA^*(I-J_{\beta_n}^{B_2})Ay_n-x_n\|\\ &\le \lim_{n\to\infty}\|y_n-x_n\| + \lim_{n\to\infty}\lambda_n\|A^*(I-J_{\beta_n}^{B_2})Ay_n\|\\ &= 0.\end{aligned} \tag{3.23}$$

From (3.17), we have

$$\lim_{n\to\infty}\|x_{n+1}-y_n+\lambda_nA^*(I-J_{\beta_n}^{B_2})Ay_n\| = 0. \tag{3.24}$$

In view of (3.20), one obtains $\lim_{n\to\infty}\lambda_n\|A^*(I-J_{\beta_n}^{B_2})Ay_n\| = 0$, which together with (3.24) implies $\lim_{n\to\infty}\|x_{n+1}-y_n\| = 0$. Note that (3.23) further yields

$$\lim_{n\to\infty}\|y_n-J_{\beta_n}^{B_1}x_n\| \le \lim_{n\to\infty}\|y_n-x_{n+1}\| + \lim_{n\to\infty}\|x_{n+1}-J_{\beta_n}^{B_1}x_n\| = 0.$$

This together with (3.22) implies

$$\lim_{n\to\infty}\|x_n-J_{\beta_n}^{B_1}x_n\| \le \lim_{n\to\infty}\|x_n-y_n\| + \lim_{n\to\infty}\|y_n-J_{\beta_n}^{B_1}x_n\| = 0. \tag{3.25}$$

From Lemma 2.3 (iii), we obtain $\lim_{n\to\infty}\|x_n-J_{\beta}^{B_1}x_n\| \le \lim_{n\to\infty}\|x_n-J_{\beta_n}^{B_1}x_n\| = 0$. From (3.19) and Lemma 2.3 (iii), we have

$$\lim_{n\to\infty}\|Aw_n-J_{\beta}^{B_2}Aw_n\| \le \lim_{n\to\infty}\|Aw_n-J_{\beta_n}^{B_2}Aw_n\| = 0.$$

It follows from (3.21) that

$$\begin{aligned}
& \lim_{n\to\infty} \|Ax_n - J_\beta^{B_2} Ax_n\| \\
\leq\ & \lim_{n\to\infty} \|Ax_n - J_\beta^{B_2} Ax_n - Aw_n + J_\beta^{B_2} Aw_n\| + \lim_{n\to\infty} \|Aw_n - J_\beta^{B_2} Aw_n\| \\
=\ & \lim_{n\to\infty} \|(I - J_\beta^{B_2})Ax_n - (I - J_\beta^{B_2})Aw_n\| + \lim_{n\to\infty} \|Aw_n - J_\beta^{B_2} Aw_n\| \\
\leq\ & \lim_{n\to\infty} \|Ax_n - Aw_n\| + \lim_{n\to\infty} \|Aw_n - J_\beta^{B_2} Aw_n\| \\
\leq\ & \lim_{n\to\infty} \|A\|\|x_n - w_n\| + \lim_{n\to\infty} \|Aw_n - J_\beta^{B_2} Aw_n\| \\
=\ & 0. \qquad (3.26)
\end{aligned}$$

Let $\hat{x}$ be a weak cluster point of $\{x_n\}$. Then, there exists a subsequence $\{x_{n_k}\}$ of $\{x_n\}$ such that $x_{n_k} \rightharpoonup \hat{x} \in H_1$. Since A is a bounded linear operator, we have $Ax_{n_k} \rightharpoonup A\hat{x}$. By (3.25), (3.26), Lemma 2.1 and Lemma 2.3 (ii), we have $\hat{x} \in \Omega$. From Lemma 2.7, we can conclude that the sequence $\{x_n\}$ converges weakly to a point in Ω. This completes the proof. □

4. Numerical Experiments

In this section, we present numerical experiments to show the efficiency of Algorithm 3.1 and compare with algorithms of Dey [11] and Zhang and Wang [29]. All numerical experiments were conducted on MacBook Pro M1 with ram 8 GB and were implemented in Matlab R2022b.

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

We set the parameters for Algorithm 3.1: $\rho_n = 0.01$, $\delta = 0.01$, and $\alpha_n = -\frac{1}{10(n+1)}$.

For Algorithm 3.1 in [29], we set the parameters: $\gamma = 0.3$ and $\lambda_n = \frac{n}{(2n+1)L}$.

For Algorithm 1 in [11], we set the parameters: $\gamma = 0.3$, $\lambda_n = \frac{1}{2L}$, $\alpha = 0.3$, $\tau_n = \frac{1}{n^2}$, $\eta_n = \frac{1}{5n+1}$, and $\theta_n = 0.8 - \eta_n$.

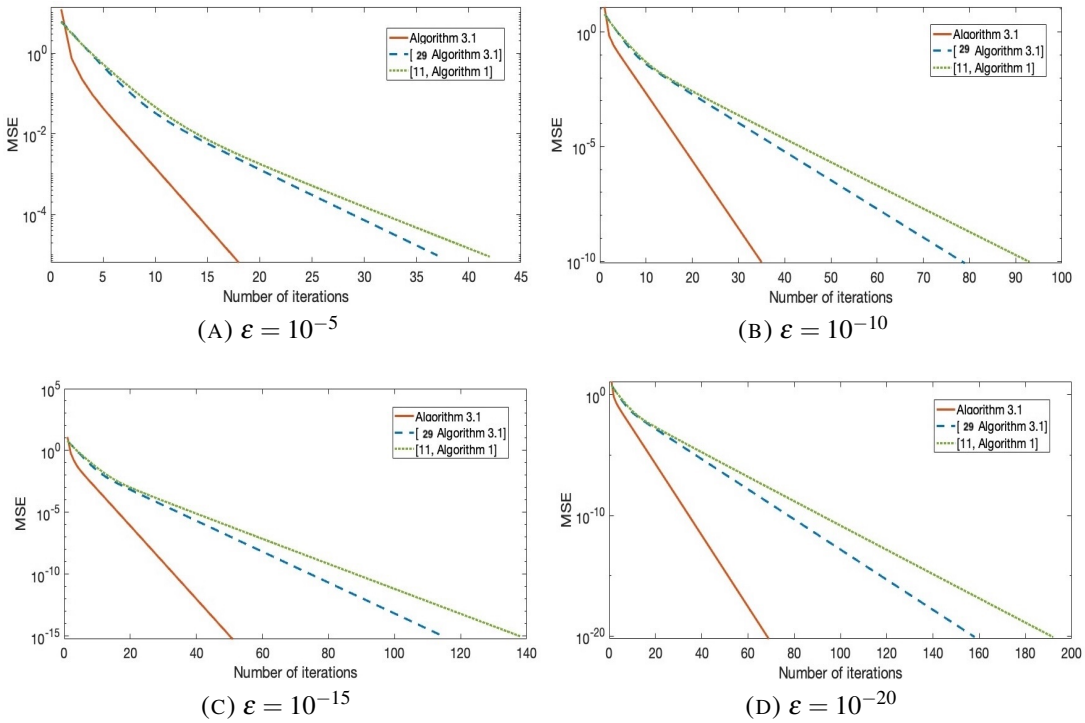
Setting MSE = $\|x_{n+1} - x_n\| \leq \varepsilon$ as the stop criterion, we obtain the results with the number of iterations, CPU time and different ε in Table 1.

TABLE 1. Comparison results on Algorithm 3.1 and the algorithms of Dey [11] and Zhang and Wang [29] for different ε with $k = 500$

ε	Algorithm 3.1		[29, Algorithm 3.1]		[11, Algorithm 1]	
	Iterations	CPU time	Iterations	CPU time	Iterations	CPU time
10^{-5}	18	0.1496	37	0.0996	42	0.1136
10^{-10}	35	0.1631	79	0.2242	93	0.2207
10^{-15}	51	0.2183	115	0.2914	138	0.3198
10^{-20}	69	0.1773	158	0.3353	192	0.6059

In Table 1, for different ε, we see that Algorithm 3.1 has the number of iterations and CPU time less than Algorithms of Dey [11] and Zhang and Wang [29]. Therefore, it can be concluded that Algorithm 3.1 performs better than other algorithms for each ε.

Next, we show the graphs of MSE plotting for each ε.



(A) $\varepsilon = 10^{-5}$ (B) $\varepsilon = 10^{-10}$

(C) $\varepsilon = 10^{-15}$ (D) $\varepsilon = 10^{-20}$

FIGURE 1. Graph of MSE for each algorithm with $k = 500$

5. APPLICATIONS

In this section, we present applications in image recovery and data classification. In the first part, we consider special cases of the split variational inclusion problem to the split feasibility problem. The second part presents image deblurring and the third part presents the hypertension prediction in data classification problems.

5.1. **Application to the split feasibility problem.** Consider the following split feasibility problem (SFP) suggested by Censor and Elfving [6]

$$\text{find } x^* \in C \text{ such that } Ax^* \in Q, \tag{5.1}$$

where C and Q are convex and closed subsets of real Hilbert spaces H_1 and H_2, respectively, and $A : H_1 \to H_2$ is a linear and bounded operator.

The definition of the subdifferential ∂g for a proper lower semicontinuous convex function $g : H \to (-\infty, \infty)$ is defined by $\partial g(x) = \{z \in H : g(x) - g(y) \leq \langle z, x - y \rangle, \ \forall y \in H\}$ for all $x \in H$.

Let C be a nonempty, convex, and closed subset of 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

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

It is known that i_C is a proper, lower semicontinuous, and convex function on H. Hence, the subdifferential ∂i_C is a maximal monotone operator. Thus the resolvent, $J_\beta^{\partial i_C}$ for each $\beta > 0$, can be defined as $J_\beta^{\partial i_C}(x) = (I+\beta\partial i_C)^{-1}(x)$ for all $x \in H$. It is known that, for any $x \in C$, $\partial i_C x = N_C x$. Putting $y = J_\beta^{\partial i_C} x = (I+\beta\partial i_C)^{-1}(x)$, $\beta > 0$, we obtain

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

Consequently, we have the following algorithm.

Algorithm 5.1. *Choose $\rho_n \in (0,2)$ and $\alpha_n \in (-1,0]$, $\delta \in [0,1)$. Let $w_{-1}, w_{-2}, x_{-1}, x_0 \in H$ be chosen arbitrarily and $\lambda_n, \gamma_n > 0$. Set $n = 0$*

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

where

$$\lambda_n = \frac{\rho_n\|(I-P_Q)Ay_n\|^2}{\|A^*(I-P_Q)Ay_n\|^2} \text{ and } \gamma_n = \frac{\rho_n\|(I-P_Q)Aw_n\|^2}{\|A^*(I-P_Q)Aw_n\|^2}$$

5.2. **Application to image recovery.** In this section, we present numerical experiments to an image recovery and compare with the algorithm of Dong et al. [10] and the algorithm of Suantai et al. [24] where these works study the split feasibility problem. 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 constrained minimization problem:

$$\min_{x \in C} \|Ax - y\|_2. \tag{5.2}$$

Therefore, the problem (5.1) can be reduced to (5.2) when $Q = \{y\}$ and $C = [0,255]^D$. We will compare the following methods with $w_{-1}, w_{-2}, x_{-1} = (1,1,1,...,1) \in \mathbb{R}^N$ and $x_0 = (0,0,0,...,0) \in \mathbb{R}^N$. We set parameters by

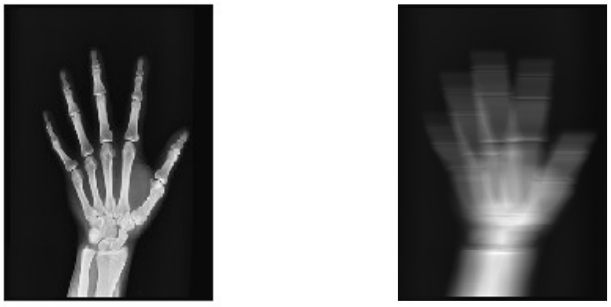
Algorithm 1 of Dong et al. [10]: $\theta = -0.9$, $\gamma = 1$, and $\alpha = 0.1$;

Algorithm 1 of Suantai et al. [24]: $\alpha_n = \min\{0.9, \frac{1}{n^{1.1}\|x_n - x_{n-1}\|^2}\}$, $\sigma = 0.5$, $\rho = 0.5$, $\mu = 0.5$, and $\gamma = 1$;

Algorithm 5.1: $\gamma = 1.9$, $\delta = 0.01$, and $\alpha_n = -\frac{-1}{10(n+1)}$.

We consider motion blur with motion length of 45 pixels and motion orientation 180°, the image size 320×266 for grey image.

To measure the restored images, we use the peak-signal-to-noise ratio (PSNR) [25] defined by $PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right)$, where MSE$= \|x_n - x\|^2$ with x being an original image. We also use structural similarity index measure (SSIM) [27] for measuring the similarity between the original and blurred images. From definitions, it is clear that the high PSNR and SSIM values show the quality of restored images. We obtain numerical results following Table 2



(A) (B)

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

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

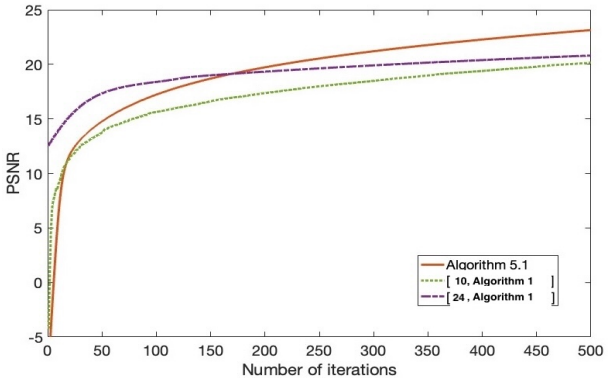
Algorithms	PSNR	SSIM	CPU time
Algorithm 5.1	23.1401	0.9002	36.7033
[10, Algorithm 1]	20.1578	0.8524	44.0284
[24, Algorithm 1]	20.8107	0.6567	25.1830

From Table 2, we observe 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.

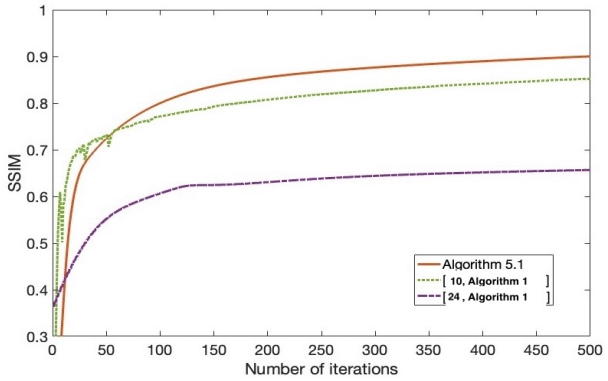
We next show the figures of restored images.

(A) PSNR=23.1401 (B) PSNR=20.1578 (C) PSNR=20.8107

FIGURE 3. (A),(B), and (C) are the restored images by Algorithm 5.1, Algorithm 1 of Dong et al. [10] and Algorithm 1 of Suantai et al. [24], 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 hypertension 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:

$$\min_{\omega \in C} \frac{1}{2}\|A\omega - \chi\|^2, \; C = \{\omega | \|\omega\|_1 \leq \xi\}, \tag{5.3}$$

where ξ is a positive constant. In particular, if $C = \{\omega | \|\omega\|_1 \leq \xi\}$ and $Q = \{\chi\}$, then (5.3) can be considered as the SFP.

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

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

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:

$$\text{Pre} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\%$$
$$\text{Rec} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%$$
$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \times 100\%$$
$$\text{F1-score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}},$$

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.

In this numerical experiments, We utilized hypertension data from the Sriphat Medical Center, Faculty of Medicine, Chiang Mai University. This dataset includes information from 15,342 patients who accessed medical services, comprising 7,671 individuals without hypertension and 7,671 individuals diagnosed with hypertension. The dataset consists of 9 attributes: Sex, Age, Temperature (Temp), Pulse, Respiratory Rate (RR), oxygen saturation (O2Sat), Body Weight, Height, and BMI, along with one outcome variable. We show visualization of hypertension dataset in Table 3.

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 $L = 20$, and the parameter is $\xi = 5$. We compare Algorithm 5.1 with other traditional machine leaning method, and the results are reported in Table 4.

In Table 4, we observe that Algorithm 5.1 performs the best accuracy, F1-score, Recall and Precision. This shows that our algorithm has the highest probability of classifying for hypertension data.

Next, we compare the performance of each algorithm for 200 iterations with Algorithm of Dong et al. [10] and Algorithm of Suantai et al. [24]. For comparison, we set $x_{-1} = x_0 = (1, 1, 1, ..., 1)$, $\theta = -0.9$, $\gamma = 0.2$ and $\alpha = 0.1$ in Algorithm 1 of Dong et al. [10].

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

Attributes	Mean	SD	CV	Min	Max
Sex	1.42	0.49	0.34	1	2
Age	59.05	14.10	0.23	21	97
Temperature	36.42	0.31	0.008	34.01	38.5
Pulse	79.51	12.03	0.15	51	109
Respiratory Rate	18.94	1.04	0.05	15	22
Oxygen saturation	97.80	0.95	0.009	96	99
Body Weight	63.17	11.59	0.18	32	91.90
Height	159.91	8.34	0.05	137.10	181.89
BMI	24.56	3.71	0.15	15.01	39.56

SD: Standard deviation; CV: Coefficient of variation.

TABLE 4. The performance in comparison of Algorithm 5.1 with traditional machine learning methods

Machine leaning method	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
Logistic regression	67.7	62.8	65.1	66.5
K neighbors	60.4	60.0	60.2	60.4
Support vector machine	70.1	58.7	63.9	66.9
Random forest	68.3	60.7	64.2	66.3
Decision tree	70.2	55.7	62.1	66.1
ELM (Algorithm 5.1)	81.3	99.9	89.6	81.3

In Algorithm 1 of Suantai et al. [24], we set $x_{-1} = (1,1,1,...,1)$, $x_0 = (0,0,0,...,0)$, $\alpha_n = \min\{0.9, \frac{1}{n^{1.1}\|x_n - x_{n-1}\|^2}\}$, $\sigma = 2$, $\rho = 0.1$, $\mu = 0.3$ and $\gamma = 0.2$.

In Algorithm 5.1, we set $w_{-1}, w_{-2}, x_{-1}, x_0 = (1,1,1,...,1)$, $\gamma = 1.8$, $\delta = 0.9$ and $\alpha_n = -\frac{-1}{10(n+1)}$.

The numerical results are reported in Table 5.

TABLE 5. The performance of each algorithm for 200 iterations

Algorithms	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
[10, Algorithm 1]	81.2099	100	89.6308	81.2099
[24, Algorithm 1]	81.2099	100	89.6308	81.2099
Algorithm 5.1	81.3514	99.9498	89.6967	81.3525

In Tables 5, we show that our algorithm obtain the best accuracy at 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. The comparison are presented in Table 6.

From Table 6, we observe that Algorithm 5.1 has more accuracy than Algorithm 1 of Dong et al. [10] and Algorithm 1 of Suantai et al. [24]. This shows that our algorithm has the highest probability of classification for the hypertension data from the Sriphat Medical Center, Faculty of Medicine, Chiang Mai University compared to other algorithms.

TABLE 6. The performance of each algorithm

Algorithms	Accuracy (%)
[10, Algorithm 1]	81.2099
[24, Algorithm 1]	81.3728
Algorithm 5.1	81.3830

6. CONCLUSIONS

In this study, we introduced a novel splitting algorithm that effectively solves the split variational inclusion problem by incorporating inertial and correction terms. The proposed method accelerates convergence while ensuring weak convergence under some conditions. Through the numerical experiments, we demonstrated that our algorithm outperforms existing methods. Moreover, the applications to the split feasibility problem provided promising results, particularly in real-world tasks such as image recovery and data classification.

Acknowledgments
This research was partially supported by Chiang Mai University. The first author was funded by National Research Council of Thailand (NRCT) under Grant No. N42A680041. Prasit Cholamjiak was supported by University of Phayao and Thailand Science Research and Innovation Fund, Thailand (Fundamental Fund 2025, Grant No. 5012/2567). We would like to thank the Sriphat Medical Center, Faculty of Medicine, Chiang Mai University for providing valuable hypertension datasets.

REFERENCES

[1] F. Alvarez, Weak convergence of a relaxed and inertial hybrid projection-proximal point algorithm for maximal monotone operators in Hilbert space, SIAM J. Optim. 14 (2004) 773-782.
[2] A. Bnouhachem, A self-adaptive iterative method for a split equality problem, Appl. Set-Valued Anal. Optim. 6 (2024) 217-231.
[3] H.H. Bauschke, P.L. Combettes, Convex Analysis and Monotone Operator Theory in Hilbert Spaces, Springer, New York, 2011.
[4] C. Byrne, Y. Censor, A. Gibali, Weak and strong convergence of algorithms for the split common null point problem, J Nonlinear Convex Anal. 13 (2011) 759-775.
[5] H. Cui, F. Wang, The split common fixed point problem with multiple output sets for strictly pseudocontractive mappings, J. Nonlinear Var. Anal. 7 (2023) 945-956.
[6] Y. Censor, T. Elfving, A multiprojection algorithm using Bregman projections in a product space, Numer Algor. 8 (1994) 221-239.
[7] Y. Censor, A. Gibali, S. Reich, The split variational inequality problem. The Technion-Israel Institue of Technology, Haifa, 2010.
[8] C.S. Chuang, Strong convergence theorems for the split variational inclusion problem in Hilbert spaces. Fixed Point Theory Appl. 2013 (2013) 350.
[9] L.C. Ceng, On inertial subgradient extragradient rule for monotone bilevel equilibrium problems, Fixed Point Theory 24 (2023) 101-126.
[10] Q.L. Dong, L. Liu, X. Qin, J.C. Yao, An alternated inertial general splitting method with linearization for the split feasibility problem, Optimization, 72 (2023) 2585-2607.
[11] S. Dey, A hybrid inertial and contraction proximal point algorithm for monotone variational inclusions, Numer. Algor. 93 (2023) 1-25.
[12] K. Goebel, W.A. Kirk, Topics in Metric Fixed Point theory. Cambridge: Cambridge University Press, 1990.

[13] A. Hanjing, S. Suantai, A fast image restoration algorithm based on a fixed point and optimization method, Mathematics, 8 (2020) 378.
[14] P. Inkrong, P. Cholamjiak, K. Wattanawitoon, U. Witthayarat, Strong convergence theorem of Bregman algorithm for solving variational inequalities in Banach spaces.,Thai J. Math. 22 (2024) 397-409.
[15] C. Izuchukwu, Y. Shehu, J.C. Yao, Convergence results for proximal point algorithm with inertial and correction terms, Appl. Anal. 104 (2025) 1732-1752.
[16] S. Kesornprom, P. Cholamjiak, Proximal type algorithms involving linesearch and inertial technique for split variational inclusion problem in hilbert spaces with applications, Optimization, 68 (2019) 2369-2395.
[17] S. Kesornprom, P. Cholamjiak, W. Cholamjiak, Strong convergence of a parallel extragradient-like algorithm involving pseudo-monotone mappings for solving common variational inequality problems, Thai J. Math. 19 (2021) 854-864.
[18] G. López, V. Martín-Márquez, H.K. Xu, Iterative algorithms for the multiple-sets split feasibility problem, In: Censor, Y, Jiang, M, Wang, G (eds.) Biomedical Mathematics: Promising Directions in Imaging, Therapy Planning and Inverse Problems, pp. 243-279. Medical Physics Publishing, Madison, 2010.
[19] A. Moudafi, Split monotone variational inclusions, J. Optim. Theory Appl. 150 (2011) 275-283.
[20] G. Marino, H.K. Xu, Convergence of generalized proximal point algorithms, Commun. Pure Appl. Anal. 3 (2004) 791-808.
[21] T.L. Nguyen, Y. Shin, Deterministic sensing matrices in compressive sensing: A survey, The Scientific World J. 2013 (2013) 192795.
[22] Z. Opial, Weak convergence of the sequence of successive approximations for nonexpansive mappings, Bull. Amer. Math. Soc. 73 (1967) 591-597.
[23] M.O. Osilike, S.C. Aniagbosor, B.G. Akuchu, Fixed points of asymptotically demicontractive mappings in arbitrary Banach spaces, Panam. Math. J. 12 (2002) 77-88.
[24] S. Suantai, B. Panyanak, S. Kesornprom, P. Cholamjiak, Inertial projection and contraction methods for split feasibility problem applied to compressed sensing and image restoration, Optimi. Lett. 16 (2022) 1725–1744.
[25] K.H. Thung, P. Raveendran, A survey of image quality measures, In: International Conference for Technical Postgraduates (TECHPOS) pp. 1-4, IEEE, 2009.
[26] B. Tan, X. Qin, On relaxed inertial projection and contraction algorithms for solving monotone inclusion problems, Adv. Comput. Math. 50 (2024) 59.
[27] B. Tan, X. Qin, J.C. Yao, Strong convergence of self-adaptive inertial algorithms for solving split variational inclusion problems with applications, J. Sci. Comput. 87 (2021) 20.
[28] X. Zuo, S. Osher and W. Li, Primal-dual damping algorithms for optimization, Ann. Math. Sci. Appl. 9 (2024) 467–504.
[29] C. Zhang, Y. Wang, Proximal algorithm for solving monotone variational inclusion, Optimization, 67 (2018) 1197-1209.