

AN ACCELERATED FORWARD-BACKWARD SPLITTING ALGORITHM FOR SOLVING INCLUSION PROBLEMS WITH APPLICATIONS TO REGRESSION AND LINK PREDICTION PROBLEMS

A. DIXIT¹, D.R. SAHU², P. GAUTAM¹, T. SOM¹, J.C. YAO^{3,4,*}

¹*Department of Mathematical Sciences, Indian Institute of Technology (BHU), Varanasi, India*

²*Department of Mathematics, Institute of Science, Banaras Hindu University, Varanasi, India*

³*Center for General Education, China Medical University, Taichung 40402, Taiwan*

⁴*Department of Applied Mathematics, National Sun Yat-sen University, Kaohsiung 804, Taiwan*

Abstract. The forward-backward method is a very popular approach to solve composite inclusion problems. In this paper, we propose a novel accelerated forward-backward algorithm to obtain the vanishing point of sum of two operators in which one is maximal monotone and other is M -cocoercive, where M is a linear bounded operator on underlying spaces. Our proposed algorithm is more general than previously known algorithms. We study the convergence behavior of proposed algorithm under mild assumptions in the framework of real Hilbert spaces. We employ our model to solve regression problems and link prediction problems for high dimensional datasets and conduct numerical experiments to support our results. This model improves convergence speed and accuracy in respective problems. We also conduct numerical experiments to support our results.

Keywords. Monotone inclusions; Convex optimization; Saddle-point problems; Regression; Link prediction.

1. INTRODUCTION

Throughout the paper, H denotes the real Hilbert space with inner product $\langle \cdot, \cdot \rangle$. This research work concerns the problem of finding the solutions of monotone inclusion problems under the environment having mild restrictions on the choice of operators and parameters in real Hilbert spaces. The monotone inclusion problem searches a point in the space at which the image of monotone operator contains zero of the space, namely

$$\text{find } x \in H \text{ such that } 0 \in T(x). \quad (1.1)$$

The problem of the form (1.1) arises in inverse problems, and several areas of applied mathematics including, signal and image processing [1, 2], game theory [3], convex optimization [4], and stochastic optimization [5, 6]. Thus, problem (1.1) serves as a blanket for many problems of fundamental importance. Forward-backward splitting method proposed by Lions and Mercier [7] is the most popular technique to solve inclusion problem (1.1). There is a vast literature on

*Corresponding author.

E-mail addresses: discover.avi92@gmail.com (A. Dixit), drsahudr@gmail.com (D.R. Sahu), pankajg.rs.mat16@iitbhu.ac.in (P. Gautam), tsom.apm@iitbhu.ac.in (T. Som), yaojc@mail.cmu.edu.tw (J.C. Yao).

Received October 5, 2020; Accepted December 13, 2020.

the study of the forward-backward method (see [8, 9, 10, 11, 12]). It assumes the operator T as sum of two monotone operators A and B . The forward-backward method for problem (1.1) is defined by the following rule: initialize with some $x_0 \in H$ and

$$x_{n+1} = (Id + \lambda_n A)^{-1}(Id - \lambda_n B)(x_n), \quad n \in \mathbb{N}, \quad (1.2)$$

where λ_n is a regularization parameter. It is known that the sequence generated by (1.2) converges weakly to solution of problem (1.1) provided that $B : H \rightarrow H$ is $1/L$ -cocoercive and $\lambda_n \in (0, 2/L)$. The forward-backward method is very popular since we can take the dense part of T into A and the ill-conditioned part into B . This algorithm is a generalization of classical proximal point algorithm and proximal gradient algorithm. Indeed, for the choice of $A = \partial g$ and $B = \nabla f$, where $f, g : H \rightarrow \mathbb{R}$ are proper lower semicontinuous convex functions, we have forward-backward algorithm is reduced to the proximal gradient algorithm ([13]):

$$x_{n+1} = (Id + \lambda_n \partial g)^{-1}(Id - \lambda_n \nabla f)(x_n), \quad n \in \mathbb{N}. \quad (1.3)$$

Proximal gradient algorithm (1.3) is weakly convergent provided that ∇f is Lipschitz continuous. Algorithm (1.3) is one-step iterative method, i.e., the new iterate only depends on its previous term. In order to accelerate the convergence rate, multi-step iterative algorithm has been proposed by using the inertial term, which is the motivation of the heavy ball method proposed by Polyak [14]. That is a discretization of second order ordinary differential equation

$$x(t)'' + \gamma x(t)' + \nabla \omega(x(t)) = 0,$$

where $x(t)$ is a time continuous trajectory, $\gamma > 0$ is a friction and $\omega(x(t))$ is an external gravitational field. To minimize a smooth function ω , the heavy ball method in its simplest form can be written by recursion as follows:

$$\begin{cases} y_n = x_n + \theta_n(x_n - x_{n-1}), \\ x_{n+1} = y_n - \lambda_n \nabla \omega(x_n), \quad n \in \mathbb{N}, \end{cases}$$

where θ_n is inertial parameter. The presence of inertial parameters greatly improves the performance of the algorithm. To minimize the smooth convex function ω with L -Lipschitz gradient, the modified form of Polyak's heavy ball method was proposed by Nesterov [15] by evaluating the gradient at extrapolated point y_n rather than x_n . It was observed in [14] that the inertial term enhances the convergence rate of the iterative algorithm from $O(1/k)$ to $O(1/k^2)$. The algorithm proposed by Nesterov [15] is given as follows:

$$\begin{cases} y_n = x_n + \theta_n(x_n - x_{n-1}), \\ x_{n+1} = y_n - \lambda_n \nabla \omega(y_n), \quad n \in \mathbb{N}, \end{cases}$$

where $\lambda_n = \frac{1}{L}$, $\theta_n = \frac{t_n - 1}{t_n + 1}$ with $t_1 = 1$ and $t_{n+1} = \frac{\sqrt{4t_n^2 + 1}}{2}$. The choice of momentum parameter θ_n generates a sequence, which behaves like $1 - \frac{3}{n}$ as $n \rightarrow \infty$.

In 2001, Alvarez and Attouch [16] took the advantage of inertial terms in the reference of proximal point algorithms. They evaluated the resolvent operator on the inertial term instead of x_n and proposed an accelerated algorithm, which is called the inertial proximal point algorithm and is given by

$$\begin{cases} y_n = x_n + \theta_n(x_n - x_{n-1}), \\ x_{n+1} = (Id + \lambda_n T)^{-1}(y_n), \quad n \in \mathbb{N}, \end{cases} \quad (1.4)$$

where $\{\lambda_n\}$ is non-decreasing and $\{\theta_n\}$ is in $[0, 1)$. Algorithm (1.4) converges to a zero of T if

$$\sum_{n=1}^{\infty} \theta_n \|x_n - x_{n-1}\|^2 < \infty. \quad (1.5)$$

In 2003, Moudafi and Oliny [17] modified the proximal point algorithm to find the zeros of the sum of a set valued operator A and a single valued operator B . The algorithm follows the following pattern:

$$\begin{cases} y_n = x_n + \theta_n(x_n - x_{n-1}), \\ x_{n+1} = (Id + \lambda_n A)^{-1}(y_n - \lambda_n Bx_n), \quad n \in \mathbb{N}. \end{cases} \quad (1.6)$$

The weak convergence of the Algorithm (1.6) is shown

shown when it satisfies summability condition (1.5) and $0 < \lambda_n < 2/L$, where L is the Lipschitz constant of B . Beck and Teboulle [18] extended Nesterov's idea to nonsmooth minimization problem, which consists of possibly a non-smooth function in the sum of A and B to solve the linear inverse problem. They proposed the fast iterative soft thresholding algorithm (FISTA) [18] to find the minima of sum of two convex function A and B in which one may be subdifferentiable. For A as a nondifferentiable function, the algorithm follows the pattern:

$$\begin{cases} y_n = x_n + \theta_n(x_n - x_{n-1}), \\ x_{n+1} = \text{prox}_{\lambda_n A}(y_n - \lambda_n \nabla B(y_n)), \quad n \in \mathbb{N} \end{cases}$$

where the choices of θ_n and λ_n are the same as in Nesterov's method.

Since last decade, an extensive literature has been published on accelerated methods [16, 18, 19, 20]. In 2015, Lorenz and Pock [21] used variable metric (or preconditioning) approach to solve monotone inclusion problem (1.1). For a bounded linear, self-adjoint and positive definite operator $M : H \rightarrow H$, the algorithm proposed by Lorenz and Pock [21] can be written as follows:

$$\begin{cases} y_n = x_n + \theta_n(x_n - x_{n-1}) \\ x_{n+1} = (Id + \lambda_n M^{-1}A)^{-1}(I - \lambda_n M^{-1}B)(y_n), \quad n \in \mathbb{N} \end{cases} \quad (1.7)$$

where $\theta_n \in [0, 1)$ is an accelerated parameter and λ_n is a step size parameter. They studied the convergence of the algorithm, which can be summarized in the following theorem.

Theorem 1.1. ([21]) *Let H be a real Hilbert space and let $A, B : H \rightarrow 2^H$ be maximally monotone operators. Further, assume that $M, L : H \rightarrow H$ are linear bounded self-adjoint and positive definite operators and that B is single-valued and cocoercive with respect to L^{-1} . Moreover, let $\lambda_n > 0$, $\theta < 1$, $\theta_n \in [0, \theta]$, $x_0 = x_1 \in H$. If*

- (i) $S_n = M - \frac{\lambda_n}{2}L$ is positive definite for all n ;
- (ii) $\sum_{n=1}^{\infty} \theta_n \|x_n - x_{n-1}\|_M^2 < \infty$,

then the sequence $\{x_n\}$ generated by Algorithm (1.7) converges weakly to a solution of the inclusion problem (1.1) with $T = A + B$.

The aim of this paper is in three folds. Our first aim is to propose a forward-backward algorithm, which converges better than Algorithm (1.7). Since second assumption in Theorem 1.1 is very strong, which is not easy to verify and reduces its practical applicability, so our second aim is to study the convergence behavior of proposed algorithm with mild assumption so that it can be helpful in practical applicability. Lastly, we aim to show the application of the proposed algorithm to regression problems and link prediction problems.

The organization of the paper is as follows. In upcoming section, Section 2, we discuss the basic definitions and the results to study the convergence behavior of the proposed algorithm. In Section 3, we propose a preconditioned forward-backward algorithm and study its convergence behavior under mild restrictions on operators and parameters. We also discuss a numerical example in the support of our findings which show that in same environment the proposed algorithm has better convergence speed than the algorithm proposed by Lorenz and Pock [21]. In Section 4, we apply the proposed algorithm to solve the saddle point problem. In the last section, Section 5, we perform numerical experiments to show the practicability of the proposed algorithm and compared its convergence speed with already known algorithms. We apply the proposed algorithm to solve regression problems and link prediction problems.

2. PRELIMINARIES

This section is devoted to some important definitions and results from nonlinear analysis and operator theory. Throughout the paper, the strong convergence of a sequence $\{x_n\}$ is denoted by \rightarrow , and the weak convergence by \rightharpoonup . Symbol \mathbb{N} and \mathbb{R}_∞ are used to denote the set of natural numbers and set of real number with ∞ , respectively. Symbol ‘ Id ’ refers to the identity map on H . $\mathcal{G}(A)$ is used to denote the graph of the set-valued operator $A : H \rightarrow 2^H$, which is defined as $\mathcal{G}(A) = \{(x, Ax) : x \in \mathcal{D}(A)\}$, where $\mathcal{D}(A)$ denotes the domain of A .

Definition 2.1. Let D be a nonempty subset of H and let $T : D \rightarrow H$ be an operator. Then T is said to be

- (i) κ -Lipschitz continuous on H if there exists $\kappa > 0$ such that

$$\|Tx_1 - Tx_2\| \leq \kappa \|x_1 - x_2\| \text{ for all } x_1, x_2 \in D;$$

- (ii) nonexpansive if

$$\|Tx_1 - Tx_2\| \leq \|x_1 - x_2\| \text{ for all } x_1, x_2 \in D;$$

- (iii) firmly nonexpansive if

$$\|Tx_1 - Tx_2\|^2 \leq \langle x_1 - x_2, Tx_1 - Tx_2 \rangle \text{ for all } x_1, x_2 \in D.$$

Definition 2.2. Let $T : H \rightarrow 2^H$ be a set-valued operator. Then T is said to monotone if, $\forall x_1, x_2 \in H$ and $u_1 \in T(x_1), u_2 \in T(x_2)$,

$$0 \leq \langle x_1 - x_2, u_1 - u_2 \rangle.$$

T is said to be maximally monotone if there exists no monotone operator $S : H \rightarrow 2^H$ such that $\mathcal{G}(S)$ properly contains $\mathcal{G}(T)$, i.e., for every $(x_1, u_1) \in H \times H$,

$$(x_1, u_1) \in \mathcal{G}(T) \Leftrightarrow \langle x_1 - x_2, u_1 - u_2 \rangle \geq 0, \forall (x_2, u_2) \in \mathcal{G}(T).$$

Let M be a bounded linear operator on H . M is said to be self-adjoint if $M^* = M$, where M^* denotes the adjoint of operator M . A self-adjoint operator M on H is said to be positive definite if $\langle M(x), x \rangle > 0$ for every nonzero $x \in H$ ([22]). Define the M -inner product $\langle \cdot, \cdot \rangle_M$ on H by $\langle x, y \rangle_M = \langle x, M(y) \rangle$ for all $x, y \in H$. The corresponding M -norm is defined by $\|x\|_M^2 = \langle x, Mx \rangle$ for all $x \in H$.

Definition 2.3. Let D be a nonempty subset of H . Let $T : D \rightarrow H$ be an operator and let $M : H \rightarrow H$ be a positive definite operator. Then T is said to be

(i) nonexpansive with respect to M -norm if

$$\|Tx_1 - Tx_2\|_M \leq \|x_1 - x_2\|_M \quad \forall x_1, x_2 \in H;$$

(ii) M -cocoercive if

$$\|Tx_1 - Tx_2\|_{M^{-1}}^2 \leq \langle x_1 - x_2, Bx_1 - Bx_2 \rangle, \text{ for all } x_1, x_2 \in H.$$

Example 2.1. Define $B : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ by $(x_1, x_2, x_3) \mapsto (5x_1, 4 \sin x_2, \tan^{-1}(5x_3))$ and

$$M = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 5 \end{bmatrix}.$$

For $x = (x_1, x_2, x_3)$ and $y = (y_1, y_2, y_3)$, we have

$$\begin{aligned} \|Bx - By\|_{M^{-1}}^2 &= 5(x_1 - y_1)^2 + 4(\sin x_2 - \sin y_2)^2 + (\tan^{-1} 5x_3 - \tan^{-1} 5y_3)^2 \\ &= \langle x - y, Bx - By \rangle. \end{aligned}$$

Thus, B is M -cocoerceive.

Lemma 2.1. ([13]) *The following identities hold, $\forall x_1, x_2 \in H$ and $a \in (0, 1)$:*

- (i) $\|x_1 - x_2\|^2 = \|x_1\|^2 + \|x_2\|^2 - 2\langle x_1, x_2 \rangle$,
- (ii) $\|ax_1 + (1-a)x_2\|^2 = a\|x_1\|^2 + (1-a)\|x_2\|^2 - a(1-a)\|x_1 - x_2\|^2$.

Lemma 2.2. ([23]) *Let $\alpha, \rho \in \mathbb{R}$ and $\rho > 0$. Then, for nonnegative α , we have*

$$\|x_1 \pm \alpha x_2\|^2 \geq (1 - \alpha\rho)\|x_1\|^2 + \alpha\left(\alpha - \frac{1}{\rho}\right)\|x_2\|^2, \text{ for all } x_1, x_2 \in H.$$

Lemma 2.3. ([24]) *Let $\{\phi_n\}, \{\zeta_n\}$ and $\{\alpha_n\}$ be sequences in $[0, \infty)$ such that*

$$\phi_{n+1} \leq \phi_n + \alpha_n(\phi_n - \phi_{n-1}) + \zeta_n \text{ for all } n \in \mathbb{N}, \quad \sum_{n=1}^{\infty} \zeta_n < \infty$$

and there exists a real number α with $0 \leq \alpha_n \leq \alpha < 1$ for all $n \in \mathbb{N}$. Then

- (i) $\sum_{n=1}^{\infty} [\phi_n - \phi_{n-1}]_+ < \infty$, where $[t]_+ = \max\{t, 0\}$,
- (ii) *there exists $\phi^* \in [0, \infty)$ such that $\phi_n \rightarrow \phi^*$.*

Lemma 2.4. ([13]) *Let C be a nonempty closed convex subset of H and let $T : C \rightarrow H$ be a nonexpansive mapping. Let $\{v_n\}$ be a sequence in C and $v \in H$ such that $v_n \rightharpoonup v$ and $v_n - Tv_n \rightarrow 0$ as $n \rightarrow \infty$. Then $v \in \text{Fix}(T)$.*

Lemma 2.5. ([13]) *Let C be a nonempty subset of H and let $\{v_n\}$ be a sequence in H such that the following two conditions hold:*

- (i) *for all $v \in C$, $\lim_{n \rightarrow \infty} \|v_n - v\|$ exists,*
- (ii) *every sequential weak cluster point of $\{v_n\}$ is in C .*

Then the sequence $\{v_n\}$ converges weakly to a point in C .

3. MAIN RESULTS

Throughout this section, we study the problem (1.1) with $T = A + B$, where $A : H \rightarrow 2^H$ is maximal monotone operator and $B : H \rightarrow H$ is M -cocoercive. Then inclusion problem (1.1) is to find $x^* \in H$ such that

$$0 \in (A + B)(x^*). \quad (3.1)$$

Let $M : H \rightarrow H$ be a linear self-adjoint and positive definite operator. For $\lambda \in (0, \infty)$, define an operator $J_{\lambda, M}^{A, B}$ by

$$J_{\lambda, M}^{A, B} = (I + \lambda M^{-1}A)^{-1}(I - \lambda M^{-1}B). \quad (3.2)$$

We give now properties of the operator $J_{\lambda, M}^{A, B}$.

Proposition 3.1. *Let $A : H \rightarrow 2^H$ be a maximal monotone operator and let $M : H \rightarrow H$ a linear bounded self-adjoint and positive definite operator. Let $B : H \rightarrow H$ be M -cocoercive operator and $\lambda \in (0, 1]$. Then we have the following:*

- (a) $I - \lambda M^{-1}B$ is nonexpansive with respect to M -norm.
- (b) $(I + \lambda M^{-1}A)^{-1}$ is nonexpansive with respect to M -norm.
- (c) The operator $J_{\lambda, M}^{A, B}$ defined by (3.2) is nonexpansive with respect to M -norm.

Proof. (a) Define $S = I - M^{-1}B$. We show that S is nonexpansive with respect to M -norm. Let $x, y \in H$. Since B is M -cocoercive, we have

$$\|Bx - By\|_{M^{-1}}^2 \leq \langle x - y, Bx - By \rangle \leq 2\langle x - y, Bx - By \rangle.$$

Thus,

$$\|M^{-1}(Bx - By)\|_M^2 = \|Bx - By\|_{M^{-1}}^2 \leq 2\langle x - y, Bx - By \rangle,$$

which implies that

$$\lambda^2 \|M^{-1}(Bx - By)\|_M^2 \leq 2\langle x - y, \lambda M^{-1}(Bx - By) \rangle_M.$$

Hence

$$\|x - y\|_M^2 + \|\lambda M^{-1}(Bx - By)\|_M^2 - 2\langle x - y, \lambda M^{-1}(Bx - By) \rangle_M \leq \|x - y\|_M^2,$$

i.e.,

$$\|(I - \lambda M^{-1}B)x - (I - \lambda M^{-1}B)y\|_M^2 \leq \|x - y\|_M^2.$$

Therefore, S is nonexpansive with respect to M -norm.

- (b) Define $T = (I + \lambda M^{-1}A)^{-1}$. Note that

$$T = (I + \lambda M^{-1}A)^{-1} \Leftrightarrow T^{-1} - I = \lambda M^{-1}A \Leftrightarrow \lambda A = M(T^{-1} - I).$$

Let $x, y \in X$. Then $M(x - T(x)) \in \lambda A(T(x))$ and $M(y - T(y)) \in \lambda A(T(y))$. Since A is monotone, we have

$$\langle Tx - Ty, M(x - Tx) - M(y - Ty) \rangle \geq 0,$$

i.e.,

$$\langle Tx - Ty, M(x - y) \rangle \geq \langle Tx - Ty, M(Tx - Ty) \rangle.$$

Then

$$\langle Tx - Ty, (x - y) \rangle_M \geq \|Tx - Ty\|_M^2.$$

Thus, T is nonexpansive with respect to M -norm.

(c) From (a) and (b), we see that the operator $T \circ S = J_{\lambda, M}^{A, B}$ is nonexpansive with respect to M -norm. \square

Proposition 3.2. *Let $A : H \rightarrow 2^H$ be a maximal monotone operator. Let $M : H \rightarrow H$ be a linear bounded self-adjoint and positive definite operator, and let $B : H \rightarrow H$ be an M -cocoercive operator. Let $\lambda \in (0, \infty)$. Then $x^* \in H$ is a solution of inclusion problem (3.1) if and only if x^* is the fixed point of the operator $J_{\lambda, M}^{A, B}$.*

Proof. Suppose that $0 \in (A + B)(x^*)$. Then

$$\begin{aligned} 0 \in \lambda A(x^*) + \lambda B(x^*) &\Leftrightarrow 0 \in \lambda M^{-1}A(x^*) + \lambda M^{-1}B(x^*) \\ &\Leftrightarrow -\lambda M^{-1}B(x^*) \in \lambda M^{-1}A(x^*) \\ &\Leftrightarrow x^* - \lambda M^{-1}B(x^*) \in x^* + \lambda M^{-1}A(x^*) \\ &\Leftrightarrow x^* = (I + \lambda M^{-1}A)^{-1}(I - \lambda M^{-1}B)(x^*). \end{aligned}$$

\square

From Proposition 3.2, we conclude that inclusion problem (3.1) can be solved by finding fixed points of the operator $J_{\lambda, M}^{A, B}$. In the light of this fact and motivated by [23], we propose the following algorithm.

Algorithm 3.1. Let $A : H \rightarrow 2^H$ be a maximal monotone operator, $M : H \rightarrow H$ a positive definite operator and $B : H \rightarrow H$ an M -cocoercive operator. Let $x_0, x_1 \in H$. The accelerated preconditioning forward-backward normal S-iteration method (APFBNSM) is defined as follows:

$$\begin{cases} y_n = x_n + \theta_n(x_n - x_{n-1}) \\ x_{n+1} = J_{\lambda, M}^{A, B}[(1 - \beta_n)y_n + \beta_n J_{\lambda, M}^{A, B}(y_n)], \end{cases} \quad \text{for all } n \in \mathbb{N}, \quad (3.3)$$

where $\beta_n \in (0, 1)$, $\lambda \in (0, 1]$, $\theta_n \in [0, 1]$.

For $\theta_n = 0$, $\forall n \in \mathbb{N}$ and $M = Id$, the accelerated preconditioning forward-backward normal S-iteration method (3.3) is reduced to the normal S-iteration based method forward-backward splitting algorithm (nS-FBSA) ([25]):

$$\begin{cases} y_n = x_n + \theta_n(x_n - x_{n-1}) \\ x_{n+1} = J_{\lambda}^{A, B}[(1 - \beta_n)y_n + \beta_n J_{\lambda}^{A, B}(y_n)], \end{cases} \quad \text{for all } n \in \mathbb{N}.$$

Assumption 3.1.

Consider the parameters $\theta_n, \beta_n, \lambda$, which satisfying the following conditions:

- (B1) $\{\theta_n\} \subset [0, \theta]$ is a non-decreasing sequence with $\theta \in [0, 1]$;
- (B2) $\{\beta_n\} \subset (0, 1)$ and $\lambda \in (0, 1]$;
- (B3) constants $\beta, \tau, \delta > 0$ satisfying

$$\delta > \frac{2\gamma\theta(\theta(1 + \theta) + \tau)}{1 - \theta^2(1 - \beta)} \quad \text{and} \quad 0 < \beta \leq \beta_n \leq \frac{\delta - \theta(2\gamma\theta(1 + \theta) + \theta\delta(1 - \beta) + 2\gamma\tau)}{\delta[1 + 2\gamma\theta(1 + \theta) + \theta\delta(1 - \beta) + 2\gamma\tau]},$$

where $\gamma = 1 + \frac{1}{\beta^2}$.

3.1. Convergence analysis of the APFBNSM.

Proposition 3.3. *Let $M : H \rightarrow H$ be a linear bounded self-adjoint and positive definite operator. Let $A : H \rightarrow 2^H$ be a maximally monotone and let $B : H \rightarrow H$ be M -cocoercive operator such that $(A + B)^{-1}(0)$ is nonempty. Assume that λ and sequences $\{\theta_n\}$ and $\{\beta_n\}$ satisfy Assumption 3.1. Let $x^* \in (A + B)^{-1}(0)$ and let $\{x_n\}$ be a sequence in H generated by Algorithm 3.1. Then*

$$\begin{aligned} \|x_{n+1} - x^*\|_M^2 &\leq (1 + \theta_n)\|x_n - x^*\|_M^2 - \theta_n\|x_{n-1} - x^*\|_M^2 + \theta_n(1 + \theta_n)\|x_n - x_{n-1}\|_M^2 \\ &\quad - \beta_n(1 - \beta_n)\|y_n - J_{\lambda, M}^{A, B}(y_n)\|_M^2 \text{ for all } n \in \mathbb{N}. \end{aligned}$$

Proof. From Algorithm 3.1 and Lemma 2.1, we have

$$\begin{aligned} &\|x_{n+1} - x^*\|_M^2 \tag{3.4} \\ &= \|J_{\lambda, M}^{A, B}[(1 - \beta_n)y_n + \beta_n J_{\lambda, M}^{A, B}(y_n)] - x^*\|_M^2 \end{aligned}$$

$$\begin{aligned} &\leq \|(1 - \beta_n)y_n + \beta_n J_{\lambda, M}^{A, B}(y_n) - x^*\|_M^2 \\ &= (1 - \beta_n)\|y_n - x^*\|_M^2 + \beta_n\|J_{\lambda, M}^{A, B}(y_n) - x^*\|_M^2 - \beta_n(1 - \beta_n)\|y_n - J_{\lambda, M}^{A, B}(y_n)\|_M^2 \\ &\leq (1 - \beta_n)\|y_n - x^*\|_M^2 + \beta_n\|y_n - x^*\|_M^2 - \beta_n(1 - \beta_n)\|y_n - J_{\lambda, M}^{A, B}(y_n)\|_M^2 \\ &= \|y_n - x^*\|_M^2 - \beta_n(1 - \beta_n)\|y_n - J_{\lambda, M}^{A, B}(y_n)\|_M^2. \end{aligned} \tag{3.5}$$

Again, from (3.1) and Lemma 2.1, we have

$$\begin{aligned} \|y_n - x^*\|_M^2 &= \|x_n + \theta_n(x_n - x_{n-1}) - x^*\|_M^2 \\ &= \|(1 + \theta_n)(x_n - x^*) - \theta_n(x_{n-1} - x^*)\|_M^2 \\ &= (1 + \theta_n)\|x_n - x^*\|_M^2 - \theta_n\|x_{n-1} - x^*\|_M^2 + \theta_n(1 + \theta_n)\|x_n - x_{n-1}\|_M^2. \end{aligned} \tag{3.6}$$

Combining (3.4) and (3.6), we have

$$\begin{aligned} \|x_{n+1} - x^*\|_M^2 &\leq (1 + \theta_n)\|x_n - x^*\|_M^2 - \theta_n\|x_{n-1} - x^*\|_M^2 + \theta_n(1 + \theta_n)\|x_n - x_{n-1}\|_M^2 \\ &\quad - \beta_n(1 - \beta_n)\|y_n - J_{\lambda, M}^{A, B}(y_n)\|_M^2. \end{aligned}$$

This completes the proof. \square

Define sequences $\{\mu_n\}$ and $\{\xi_n\}$ by

$$\mu_n = \theta_n(1 + \theta_n) + \frac{\theta_n(1 - \beta_n)(1 - \theta_n\rho_n)}{2\gamma\beta_n\rho_n} \text{ and } \xi_n = -\frac{(1 - \beta_n)(\theta_n\rho_n - 1)}{2\gamma\beta_n}, \tag{3.7}$$

where $\rho_n = \frac{1}{\theta_n + \delta\beta_n}$.

Proposition 3.4. *Let $M : H \rightarrow H$ be a linear bounded self-adjoint and positive definite operator. Let $A : H \rightarrow 2^H$ be a maximally monotone and let $B : H \rightarrow H$ be M -cocoercive operator such*

that $(A+B)^{-1}(0)$ is nonempty. Assume that λ and sequences $\{\theta_n\}$ and $\{\beta_n\}$ satisfy Assumption 3.1. Let $x^* \in (A+B)^{-1}(0)$ and $\{x_n\}$ be a sequence in H generated by Algorithm 3.1. Then

$$\phi_{n+1} - (1 + \theta_n)\phi_n + \theta_n\phi_{n-1} \leq \xi_n \|x_{n+1} - x_n\|_M^2 + \mu_n \|x_n - x_{n-1}\|_M^2, n \in \mathbb{N}, \quad (3.8)$$

where $\phi_n = \|x_n - x^*\|_M^2$.

Proof. Set $z_n = (1 - \beta_n)y_n + \beta_n J_{\lambda, M}^{A, B}(y_n)$. Then Algorithm 3.1 can be written as:

$$\begin{cases} y_n = x_n + \theta_n(x_n - x_{n-1}) \\ z_n = (1 - \beta_n)y_n + \beta_n J_{\lambda, M}^{A, B}(y_n) \\ x_{n+1} = J_{\lambda, M}^{A, B}(z_n). \end{cases} \quad (3.9)$$

From (3.9), we have

$$\begin{aligned} \|y_n - J_{\lambda, M}^{A, B}(y_n)\|_M^2 &= \frac{1}{\beta_n^2} \|z_n - y_n\|_M^2 \\ &\geq \frac{1}{\beta_n^2} \|J_{\lambda, M}^{A, B}(z_n) - J_{\lambda, M}^{A, B}(y_n)\|_M^2 \\ &= \frac{1}{\beta_n^2} \|x_{n+1} - J_{\lambda, M}^{A, B}(y_n)\|_M^2 \\ &= \frac{1}{\beta_n^2} \|x_{n+1} - y_n + y_n - J_{\lambda, M}^{A, B}(y_n)\|_M^2. \end{aligned}$$

Taking $\rho = \frac{1}{2}$ and using Lemma 2.2, we obtain

$$\|y_n - J_{\lambda, M}^{A, B}(y_n)\|_M^2 \geq \frac{1}{\beta_n^2} \left\{ \frac{1}{2} \|x_{n+1} - y_n\|_M^2 - \|y_n - J_{\lambda, M}^{A, B}(y_n)\|_M^2 \right\},$$

which implies that

$$\left(1 + \frac{1}{\beta_n^2}\right) \|y_n - J_{\lambda, M}^{A, B}(y_n)\|_M^2 \geq \frac{1}{2\beta_n^2} \|x_{n+1} - y_n\|_M^2.$$

Since β_n is bounded below by β , we have

$$\begin{aligned} \left(1 + \frac{1}{\beta^2}\right) \|y_n - J_{\lambda, M}^{A, B}(y_n)\|_M^2 &\geq \frac{1}{2\beta_n^2} \|x_{n+1} - y_n\|_M^2 \\ &= \frac{1}{2\beta_n^2} \|x_{n+1} - x_n - \theta_n(x_n - x_{n-1})\|_M^2. \end{aligned}$$

Again, using Lemma 2.2, we obtain

$$\begin{aligned} &\left(1 + \frac{1}{\beta^2}\right) \|y_n - J_{\lambda, M}^{A, B}(y_n)\|_M^2 \quad (3.10) \\ &\geq \frac{(1 - \theta_n \rho_n)}{2\beta_n^2} \|x_{n+1} - x_n\|_M^2 + \frac{\theta_n}{2\beta_n^2} \left(\theta_n - \frac{1}{\rho_n}\right) \|x_n - x_{n-1}\|_M^2 \\ &= \frac{(1 - \theta_n \rho_n)}{2\beta_n^2} \|x_{n+1} - x_n\|_M^2 - \frac{\theta_n(1 - \theta_n \rho_n)}{2\beta_n^2 \rho_n} \|x_n - x_{n-1}\|_M^2. \end{aligned} \quad (3.11)$$

Multiplying (3.10) by $-\beta_n(1-\beta_n)$, we obtain

$$\begin{aligned} -\gamma\beta_n(1-\beta_n)\|y_n - J_{\lambda,M}^{A,B}(y_n)\|_M^2 &\leq -\frac{(1-\beta_n)(1-\theta_n\rho_n)}{2\beta_n}\|x_{n+1} - x_n\|_M^2 \\ &\quad + \frac{\theta_n(1-\beta_n)(1-\theta_n\rho_n)}{2\beta_n\rho_n}\|x_n - x_{n-1}\|_M^2. \end{aligned} \quad (3.12)$$

From Proposition 3.3 and (3.12), we get

$$\begin{aligned} &\|x_{n+1} - x^*\|_M^2 \\ &\leq (1+\theta_n)\|x_n - x^*\|_M^2 - \theta_n\|x_{n-1} - x^*\|_M^2 + \theta_n(1+\theta_n)\|x_n - x_{n-1}\|_M^2 \\ &\quad - \frac{(1-\beta_n)(1-\theta_n\rho_n)}{2\gamma\beta_n}\|x_{n+1} - x_n\|_M^2 + \frac{\theta_n(1-\beta_n)(1-\theta_n\rho_n)}{2\gamma\beta_n\rho_n}\|x_n - x_{n-1}\|_M^2. \end{aligned} \quad (3.13)$$

Hence

$$\begin{aligned} \phi_{n+1} &\leq (1+\theta_n)\phi_n - \theta_n\phi_{n-1} + \theta_n(1+\theta_n)\|x_n - x_{n-1}\|_M^2 \\ &\quad - \frac{(1-\beta_n)(1-\theta_n\rho_n)}{2\gamma\beta_n}\|x_{n+1} - x_n\|_M^2 + \frac{\theta_n(1-\beta_n)(1-\theta_n\rho_n)}{2\gamma\beta_n\rho_n}\|x_n - x_{n-1}\|_M^2, \end{aligned}$$

which can be written as

$$\phi_{n+1} - (1+\theta_n)\phi_n + \theta_n\phi_{n-1} \leq \xi_n\|x_{n+1} - x_n\|_M^2 + \mu_n\|x_n - x_{n-1}\|_M^2.$$

□

Proposition 3.5. *Suppose that $\{\theta_n\}$ and $\{\beta_n\}$ are sequences in $[0, 1)$ satisfying Assumption 3.1. Let $\{\xi_n\}$ and $\{\mu_n\}$ be sequences defined by (3.7). Then $\xi_n + \mu_{n+1} \leq -\tau$ for all $n \in \mathbb{N}$.*

Proof. Observe that

$$\mu_n = \theta_n(1+\theta_n) + \frac{\theta_n(1-\beta_n)(1-\theta_n\rho_n)}{2\gamma\beta_n\rho_n} > 0,$$

since $\theta_n\rho_n < 1$ and $\beta_n \in (0, 1)$. Again, taking into account of choice of ρ_n , we have

$$\delta = \frac{1-\theta_n\rho_n}{\rho_n\beta_n}.$$

Note

$$\mu_n = \theta_n(1+\theta_n) + \frac{\theta_n(1-\beta_n)\delta}{2\gamma} \leq \theta(1+\theta) + \frac{\theta\delta(1-\beta)}{2\gamma} \quad \text{for all } n \in \mathbb{N}. \quad (3.14)$$

For all $n \in \mathbb{N}$, we have

$$\begin{aligned}
\xi_n + \mu_{n+1} \leq -\tau &\Leftrightarrow \frac{(1 - \beta_n)(\theta_n \rho_n - 1)}{2\gamma\beta_n} + (\mu_{n+1} + \tau) \leq 0 \\
&\Leftrightarrow (1 - \beta_n)(\theta_n \rho_n - 1) + 2\gamma\beta_n(\mu_{n+1} + \tau) \leq 0 \\
&\Leftrightarrow -(1 - \beta_n)\delta\rho_n\beta_n + 2\gamma\beta_n(\mu_{n+1} + \tau) \leq 0 \\
&\Leftrightarrow -\frac{(1 - \beta_n)\delta}{\theta_n + \delta\beta_n} + 2\gamma(\mu_{n+1} + \tau) \leq 0 \\
&\Leftrightarrow -(1 - \beta_n)\delta + 2\gamma(\mu_{n+1} + \tau)(\theta_n + \delta\beta_n) \leq 0 \\
&\Leftrightarrow 2\gamma(\mu_{n+1} + \tau)(\theta_n + \delta\beta_n) + \beta_n\delta \leq \delta.
\end{aligned}$$

By using (3.14), we have

$$\begin{aligned}
2\gamma(\mu_{n+1} + \tau)(\theta_n + \delta\beta_n) + \beta_n\delta &\leq 2\gamma(\theta(1 + \theta) + \frac{\theta\delta(1 - \beta)}{2\gamma} + \tau)(\theta + \delta\beta_n) + \beta_n\delta \\
&\leq \delta,
\end{aligned}$$

where the last inequality follows by using the upper bound for $\{\beta_n\}$ in Assumption 3.1. Hence

$$\xi_n + \mu_{n+1} \leq -\tau \text{ for all } n \in \mathbb{N}.$$

□

Now, we establish the weak convergence of the accelerated preconditioning forward-backward normal S -iteration method (APFBNSM) defined by Algorithm 3.1 for the computation of solutions of inclusion problem (3.1).

Theorem 3.1. *Let $M : H \rightarrow H$ be a linear bounded self-adjoint and positive definite operator. Let $A : H \rightarrow 2^H$ be a maximally monotone and let $B : H \rightarrow H$ be M -cocoercive operator such that $(A + B)^{-1}(0)$ is nonempty. Let $\lambda \in (0, 1]$ and let $\{\theta_n\}$ and $\{\beta_n\}$ satisfy the Assumption 3.1 with $\theta_1 = 0$. Then the sequence $\{x_n\}$ generated by Algorithm 3.1 converges weakly to a point of $(A + B)^{-1}(0)$.*

Proof. Let $x^* \in (A + B)^{-1}(0)$. Set $\psi_n = \phi_n - \theta_n\phi_{n-1} + \mu_n\|x_n - x_{n-1}\|_M^2$. We proceed with the following steps.

Step 1. $\sum_{n=1}^{\infty} \|x_{n+1} - x_n\|_M^2 < \infty$.

Consider

$$\begin{aligned}
\psi_{n+1} - \psi_n &= \phi_{n+1} - \theta_{n+1}\phi_n + \mu_{n+1}\|x_{n+1} - x_n\|_M^2 - \phi_n + \theta_n\phi_{n-1} - \mu_n\|x_n - x_{n-1}\|_M^2 \\
&= \phi_{n+1} - (1 + \theta_{n+1})\phi_n + \theta_n\phi_{n-1} + \mu_{n+1}\|x_{n+1} - x_n\|_M^2 - \mu_n\|x_n - x_{n-1}\|_M^2.
\end{aligned}$$

Using Propositions 3.4 and 3.5, we have

$$\begin{aligned}
\psi_{n+1} - \psi_n &\leq \xi_n\|x_{n+1} - x_n\|_M^2 + \mu_{n+1}\|x_{n+1} - x_n\|_M^2 \\
&= (\xi_n + \mu_{n+1})\|x_{n+1} - x_n\|_M^2 \\
&\leq -\tau\|x_{n+1} - x_n\|_M^2 \text{ for all } n \in \mathbb{N},
\end{aligned} \tag{3.15}$$

which implies that $\{\psi_n\}$ is nonincreasing sequence. Since $\{\theta_n\}$ is bounded, we obtain

$$-\theta\phi_{n-1} \leq \phi_n - \theta\phi_{n-1} \leq \psi_n \leq \psi_1.$$

Thus,

$$\begin{aligned} \phi_n &\leq \theta\phi_{n-1} + \psi_1 \\ &\leq \theta(\theta\phi_{n-2} + \psi_1) + \psi_1 \\ &\vdots \\ &\leq \theta^n\phi_0 + \psi_1 \sum_{k=0}^{n-1} \theta^k \leq \theta^n\phi_0 + \frac{\psi_1}{1-\theta}. \end{aligned}$$

From (3.15), we conclude that

$$\begin{aligned} \tau \sum_{k=1}^n \|x_{k+1} - x_k\|_M^2 &\leq \psi_1 - \psi_{n+1} \\ &\leq \psi_1 + \theta\phi_n \\ &\leq \psi_1 + \theta\left(\theta^n\phi_0 + \frac{\psi_1}{1-\theta}\right) \\ &= \theta^{n+1}\phi_0 + \frac{\psi_1}{1-\theta}. \end{aligned}$$

Since $\theta^{n+1} \rightarrow 0$ as $n \rightarrow \infty$, we obtain that

$$\sum_{n=1}^{\infty} \|x_{n+1} - x_n\|_M^2 < \infty. \quad (3.16)$$

Step 2. $\lim_{n \rightarrow \infty} \|x_n - x^*\|_M$ exists.

From (3.8), (3.14), (3.16) and Lemma 2.3, we obtain $\lim_{n \rightarrow \infty} \|x_n - x^*\|_M$ exists.

Step 3. Every sequential weak cluster point of sequence $\{x_n\}$ is in $\text{Fix}(J_{\lambda, M}^{A, B})$.

From (3.16), we have

$$\lim_{n \rightarrow \infty} \|x_n - x_{n-1}\|_M = 0. \quad (3.17)$$

From Algorithm 3.1, we have

$$\|y_n - x_{n+1}\|_M \leq \|x_n - x_{n+1}\|_M + \theta \|x_n - x_{n-1}\|_M.$$

Using (3.17), we get that $\lim_{n \rightarrow \infty} \|y_n - x_{n+1}\|_M = 0$. From (3.3), we have

$$\begin{aligned}
\|J_{\lambda, M}^{A, B} y_n - y_n\|_M &= \|J_{\lambda, M}^{A, B} y_n - x_{n+1} + x_{n+1} - y_n\|_M \\
&\leq \|J_{\lambda, M}^{A, B} y_n - x_{n+1}\|_M + \|x_{n+1} - y_n\|_M \\
&= \|J_{\lambda, M}^{A, B} y_n - J_{\lambda, M}^{A, B} z_n\|_M + \|x_{n+1} - y_n\|_M \\
&\leq \|y_n - z_n\|_M + \|x_{n+1} - y_n\|_M \\
&= \|y_n - (1 - \beta_n)y_n - \beta_n J_{\lambda, M}^{A, B} y_n\|_M + \|x_{n+1} - y_n\|_M \\
&= \beta_n \|y_n - J_{\lambda, M}^{A, B} y_n\|_M + \|x_{n+1} - y_n\|_M,
\end{aligned}$$

which implies that

$$(1 - \beta_n) \|J_{\lambda, M}^{A, B} y_n - y_n\|_M \leq \|x_{n+1} - y_n\|_M. \quad (3.18)$$

From (3.18), we obtain

$$\|J_{\lambda, M}^{A, B} y_n - y_n\|_M \rightarrow 0 \text{ as } n \rightarrow \infty. \quad (3.19)$$

Suppose that $\{x_n\}$ has a weak cluster point $x \in H$. Then $\{x_n\}$ has a subsequence $\{x_{n_k}\}$, which converges weakly to x . From (3.19) and Lemma 2.4, we have $x \in (A + B)^{-1}(0)$. It follows from Lemma 2.5 that $\{x_n\}$ converges weakly to a point in $(A + B)^{-1}(0)$. \square

Remark 3.1. In order to deal with the convergence of Algorithm 3.1, we assume that $\theta_1 = 0$ in Theorem 3.1. We can obtain the same conclusion of Theorem 3.1 if we assume $x_1 = x_0$.

3.2. Numerical comparison of Algorithms (1.7) and 3.1. The aim of the numerical example is to study the convergence behaviour of Algorithm 3.1 to solve the inclusion problem and compare its performance with Algorithm (1.7).

Let $H = \mathbb{R}^3$ with Euclidean norm and let $A : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ be an operator defined by

$$A(x_1, x_2, x_3) = (x_2 - x_3, x_3 - x_1, x_1 - x_2) \text{ for all } (x_1, x_2, x_3) \in \mathbb{R}^3.$$

Then operator A is maximally monotone. Consider the operators B and M as in Example 2.1. The operator B is M -cocoerceive. Thus, we can apply Algorithms (1.7) and (3.1) to find the zeros of $A + B$. We choose initial points $x_1 = x_0 = (15, 15, 14)$, $\alpha_n = \frac{1}{20}$ and $\beta_n = 0.5$. We perform the experiment for 70 iterations with difference of norms of two consecutive values should be less than 0.001 as the stopping criterion. The graph is plotted between Euclidean norm of x_n and the number of iterations.

In Figure 1, we can observe that $\|x_n\|_2$ corresponding to Algorithm 3.1 goes towards 0 as number of iteration increases, which supports the result proved in Theorem 3.1. It can also observe that graph of Algorithm (1.7) also converges to 0. From Figure 1, we can say that the convergence speed of Algorithm 3.1 is faster than Algorithm (1.7). Table 1 shows that in order to get value of $\|x_n\|_2$ less than 3 decimal place, Algorithm 3.1 needs 53 iterations while Algorithm (1.7) takes just 15 iterations to obtain the same goal. This observation shows that convergence speed of Algorithm (3.1) is very faster than Algorithm (1.7), which in turn implies the importance of Algorithm 3.1 over Algorithm (1.7).

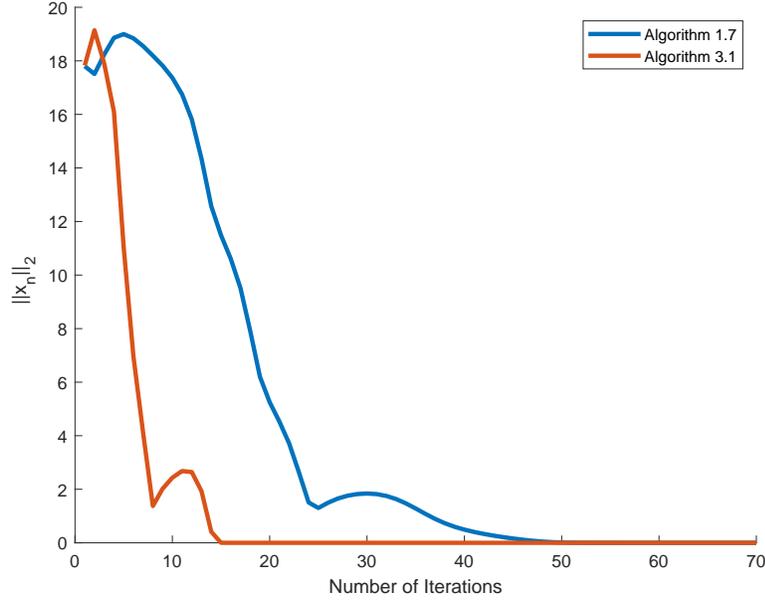


FIGURE 1. Behaviour of $\|x_n\|_2$ with respect to number of iteration.

4. APPLICATIONS

4.1. **Convex concave saddle point problem.** Consider H_1 and H_2 are two Hilbert spaces. To define the saddle point problem, we consider the following convex functions:

- $f : H_1 \rightarrow \mathbb{R}_\infty$.
- $g : H_1 \rightarrow \mathbb{R}_\infty$ is differentiable with L_g -Lipschitz gradient.
- $h^* : H_2 \rightarrow \mathbb{R}_\infty$.
- $k^* : H_2 \rightarrow \mathbb{R}_\infty$ is differentiable with L_k Lipschitz gradient.

The saddle point problem is defined as follows:

$$\min_{x \in H_1} \max_{y \in H_2} f(x) + g(x) + \langle Lx, y \rangle - h^*(y) - k^*(y), \quad (4.1)$$

where $L : H_1 \rightarrow H_2$ is a bounded linear operator. Let \mathbb{S} denote the solution set of saddle point problem (4.1).

Define operators A and B on $H_1 \times H_2$ by

$$A := \begin{bmatrix} \partial f & L^* \\ -L & \partial h^* \end{bmatrix} \text{ and } B := \begin{bmatrix} \nabla g & 0 \\ 0 & \nabla k^* \end{bmatrix}.$$

Note that A and B are maximally monotone operators. Thus using the above argument and Karush-Kuhn-Tucker condition, saddle point problem (4.1) can be formulated as the following inclusion problem

$$0 \in (A + B) \begin{bmatrix} x \\ y \end{bmatrix}.$$

Number of iteration.	Algorithm (1.7)	Algorithm 3.1
1	17.801304917618964	17.826809506966804
3	18.238459918161315	17.896347714673194
5	18.998465909761100	11.038168038343564
7	18.550509556843764	4.104266737615634
9	17.820863132624478	2.019351546352653
11	16.751374190962910	2.681677869669982
13	14.336675808934617	1.926464479485708
15	11.487166440018544	7.521568964188269e-04
17	9.510101400429985	0
19	6.199317809338941	0
21	4.524962881620460	0
23	2.637915932224955	0
25	1.302567649926143	0
27	1.653802630048002	0
29	1.820947035691583	0
31	1.819006839555166	0
37	0.892617910214436	0
39	0.598832844725429	0
41	0.405996149023377	0
43	0.268613850479600	0
45	0.161184227430202	0
50	0.012670530567700	0
52	0.002881426384463	0
53	0.001361524082377	0
54	6.428424686431453e-04	0
55	0	0
56	0	0
58	0	0
60	0	0
70	0	0

TABLE 1. The evaluation of $\|x_n\|_2$ as number of iteration increases for Algorithm (1.7) and Algorithm 3.1

We show that saddle point problem can be solved by adapting our Algorithm 3.1. For this, we consider the linear operator

$$M = \begin{bmatrix} L_g Id & 0 \\ 0 & L_k Id \end{bmatrix}.$$

The convergence analysis can be summarized in following theorem.

Theorem 4.1. *Suppose that solution set \mathbb{S} is nonempty. Let $\lambda \in (0, 1]$ and let parameters $\{\theta_n\}$ and $\{\beta_n\}$ satisfy Assumption 3.1. Let $\{(x_n, y_n)\}$ be the sequence in $H_1 \times H_2$ generated from*

initial points $(x_0, y_0) = (x_1, y_1) \in H_1 \times H_2$ and defined by

$$\begin{cases} \mu_n = x_n + \theta_n(x_n - x_{n-1}) \\ v_n = y_n + \theta_n(y_n - y_{n-1}) \\ u_n = (1 - \beta_n)\mu_n + \beta_n\xi^{-1}\{\chi_1(\mu_n) - \chi_2(v_n)\} \\ v_n = (1 - \beta_n)\mu_n + \beta_n\xi^{-1}\{\zeta_1(\mu_n) + \zeta_2(v_n)\} \\ x_{n+1} = \xi^{-1}\{\chi_1(u_n) - \chi_2(v_n)\} \\ y_{n+1} = \xi^{-1}\{\zeta_1(u_n) + \zeta_2(v_n)\}, n \in \mathbb{N}, \end{cases} \quad (4.2)$$

where $\chi_1 \equiv (L_g Id - \lambda \nabla g)(L_k Id + \lambda \partial h^*)$, $\chi_2 \equiv \lambda L^*(L_k Id - \lambda \nabla k^*)$, $\zeta_1 \equiv \lambda L(L_g Id - \lambda \nabla g)$, $\zeta_2 \equiv (\lambda \partial f + L_g Id)(L_k Id - \lambda \nabla k^*)$ and $\xi \equiv (\lambda \partial f + L_g Id)(L_k Id + \lambda \partial h^*) + \lambda^2 LL^*$. Then sequence $\{(x_n, y_n)\}$ converges weakly to a point in the solution set, \mathbb{S} .

Proof. Since g is convex with L_g -Lipschitz continuous gradient, it follows from Baillon-Hadded Theorem [13] that ∇g is cocoercive with respect to L_g^{-1} . Similarly, ∇k^* is cocoercive with respect to L_k^{-1} . For $(x, y), (\xi, \zeta) \in H_1 \times H_2$, we have

$$\begin{aligned} & \langle B(x, y) - B(\xi, \zeta), (x, y) - (\xi, \zeta) \rangle_{H_1 \times H_2} \\ &= \langle \nabla g(x) - \nabla g(\xi), x - \xi \rangle_{H_1} + \langle \nabla k^*(y) - \nabla k^*(\zeta), y - \zeta \rangle_{H_2} \\ &\geq L_g^{-1} \|\nabla g(x) - \nabla g(\xi)\|_{H_1}^2 + L_k^{-1} \|\nabla k^*(y) - \nabla k^*(\zeta)\|_{H_2}^2. \end{aligned}$$

Thus, B is M -cocoercive. With the above choice of A , B and M , Algorithm 3.1 reduces to Algorithm (4.2). Since parameters $\{\theta_n\}$ and $\{\beta_n\}$ satisfies conditions (B1), (B2) and (B3), Algorithm (4.2) converges weakly to a point in the solution set \mathbb{S} . \square

Remark 4.1. In order to solve saddle point problem (4.1), when B is M -cocoercive, the Algorithm (1.7) proposed by Lorenz and Pock [21] can be written as follows:

$$\begin{cases} \mu_n = x_n + \theta_n(x_n - x_{n-1}) \\ v_n = y_n + \theta_n(y_n - y_{n-1}) \\ x_{n+1} = \xi^{-1}\{\chi_1(\mu_n) - \chi_2(v_n)\} \\ y_{n+1} = \xi^{-1}\{\zeta_1(\mu_n) + \zeta_2(v_n)\}, n \in \mathbb{N}, \end{cases} \quad (4.3)$$

where operators $\xi, \chi_1, \chi_2, \zeta_1$ and ζ_2 are as in Theorem 4.1. If parameters θ_n and λ satisfy the assumptions as in Theorem 1.1, then the sequence $\{(x_n, y_n)\}$ generated by Algorithm (4.3) converges weakly to a point in solution set \mathbb{S} .

4.2. Lasso problem. The Lasso problem is extensively used in the field of signal processing, image processing and machine learning (see [26, 27, 28]). Many problems arising in these fields can be expressed as lasso problem. Many problems arising in these fields can be expressed as the lasso problem. For the linear operator $X : \mathbb{R}^d \rightarrow \mathbb{R}^m$ and $b \in \mathbb{R}^m$, the lasso problem is given as follows:

$$\min_{x \in \mathbb{R}^d} F(x) = \frac{1}{2m} \|Xx - b\|^2 + \rho \|x\|_1, \quad (4.4)$$

where ρ is the sparsity controlling parameter.

For the choice of $f = \rho \|x\|_1$, $g = \frac{1}{2m} \|Xx - b\|^2$ and $h^* = k^* = L = 0$, Lasso problem (4.4) is reduced to the saddle point problem (4.1). Thus, we have the following result.

Corollary 4.1. *Suppose that the solution set of Lasso problem (4.4) is nonempty. Let parameter $\{\theta_n\}$, $\{\beta_n\}$ and λ satisfy Assumption 3.1. Consider the sequence $\{x_n\}$ generated by the following algorithm with the initial point $x_0 = x_1$ and defined by,*

$$\begin{cases} \mu_n = x_n + \theta_n(x_n - x_{n-1}) \\ u_n = (1 - \beta_n)\mu_n + \beta_n(Id + \lambda L_g^{-1} \rho \partial \|\cdot\|_1)^{-1} \left(\mu_n - \frac{\lambda L_g^{-1} X^T (X\mu_n - b)}{m} \right) \\ x_{n+1} = (Id + \lambda L_g^{-1} \rho \partial \|\cdot\|_1)^{-1} \left(u_n - \frac{\lambda L_g^{-1} X^T (Xu_n - b)}{m} \right). \end{cases} \quad (4.5)$$

Then $\{x_n\}$ converges weakly to an optimal point of Lasso problem.

Proof. Using Theorem 4.1, we can obtain that Algorithm (4.5) converges weakly to an optimal point of Lasso problem (4.4). \square

Remark 4.2. Algorithm (4.3) can be used to solve Lasso problem (4.4). For the choice of $f = \rho \|x\|_1$, $g = \frac{1}{2m} \|Xx - b\|^2$ and $h^* = k^* = L = 0$, the algorithm is reduced to the following

$$\begin{cases} \mu_n = x_n + \theta_n(x_n - x_{n-1}) \\ x_{n+1} = (Id + \lambda L_g^{-1} \rho \partial \|\cdot\|_1)^{-1} \left(\mu_n - \frac{\lambda L_g^{-1} X^T (X\mu_n - b)}{m} \right). \end{cases} \quad (4.6)$$

With the assumptions as in Remark 4.1, the sequences $\{x_n\}$ generated by Algorithm 4.6 converges weakly to a solution of the Lasso problem.

5. NUMERICAL EXPERIMENT

In this section, we perform numerical experiments to demonstrate the real world applicability of the proposed algorithm. All the numerical experiments are performed in the MATLAB 2018a environment on Intel(R)core(TM)i5 processor with 8GB RAM and 64-bit operating system.

5.1. Regression problem. In this subsection, we compare the performance of Algorithms (4.2) and (4.3) for a regression problem on high dimensional datasets. The objective function we will consider is the loss function with l_1 -regularization, i.e., Lasso problem (4.4). Problems arising in different areas of science (see [29, 30, 31]) can be framed as Lasso problem (4.4). We employ both the Algorithms (4.2) and (4.3) to solve the Lasso problem (4.4) and compare their performance on the basis of their convergence speed and accuracy. For our experiment, we consider the Lasso problem with data $(X_i, b_i), i = 1, 2, \dots, m$, where $X_i = (X_{i1}, X_{i2}, \dots, X_{id})^T$ are predictor variables and b_i are responses. The description of the datasets is summarized in Table 2. Here, the total number of vertices and edges in a network is represented by the symbols $|V|$ and $|E|$, respectively. $\langle D \rangle$ represents average shortest path length, $\langle K \rangle$, the average degree, and $\langle C \rangle$, the average clustering coefficient of the network. For experimental purpose, we select the inertial parameter $\theta_n = \frac{n-1}{14n+2.5}$ and $\beta_n = 0.5 + \frac{1}{200n}$ which satisfy the conditions (B1), (B2) and (B3). A bias column is added to data set and we run algorithms for maximum 1000 iteration. We select $\theta \times \|Xb^T\|_\infty$ as a regularization parameter and vary θ between 10^{-10} to 10^{-3} in the multiple of 0.1. The best results are shown here.

TABLE 2. Topological information of real-world network datasets

Datasets	$ V $	$ E $	$\langle D \rangle$	$\langle K \rangle$	$\langle C \rangle$
Dolphin	62	159	3.302	5.129	0.258
Football	115	613	2.486	10.660	0.403
Jazz	198	2742	2.235	27.697	0.620
Celegansneural	297	2148	2.447	14.465	0.308
Usair97	332	2126	2.738	12.807	0.749
Netscience (NS)	379	914	6.042	4.823	0.798
Political blogs (PB)	1222	16714	2.738	27.355	0.360

In first experiment, we compare the performance of Algorithm (4.2) and Algorithm (4.3) on the basis of their convergence speed. We compute the difference between objective function value $F(x)$ and optimized value $F(x^*)$ at each iteration for both Algorithms (4.2) and (4.3). We initialize the experiment with point $x_0 = x_1 = 0 \in \mathbb{R}^d$. The numerical results are reported in Figure 2 for 1000 iterations.

In Figure 2, we plot the graph between $F(x) - F(x^*)$ and the number of iterations. From Figure 2, we can observe that Algorithm (4.2) has better convergence speed than Algorithm (4.3) for all datasets.

In second experiment, we compare both the Algorithms (4.2) and (4.3) on the basis of their accuracy. We calculate the root mean square error (RMSE) of Algorithm (4.2) and Algorithm (4.3) at each iteration. We take the initial points $x_0 = x_1 = 0 \in \mathbb{R}^d$ and plot the graph between RMSE and the number of iterations, which is shown in Figure 3.

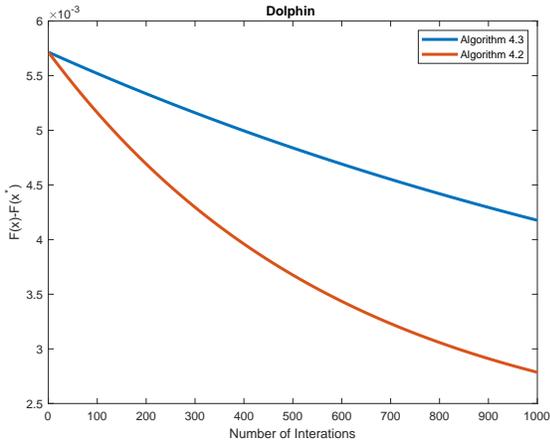
From Figure 3, we can observe that at each iteration the RMSE value of Algorithm (4.2) is less than RMSE value of Algorithm (4.3) for all datasets. Thus, Algorithm (4.2) is more accurate than Algorithm (4.3).

Remark 5.1. From Experiment 1 and Experiment 2, we observe that Algorithm (4.2) not only have higher convergence speed but it also gives more accurate results than Algorithm (4.3) for high dimensional datasets also. Thus, we observe that Algorithm (4.2) is equally important over Algorithm (4.3) for high dimensional datasets also, as we have obtained in numerical example 3.2.

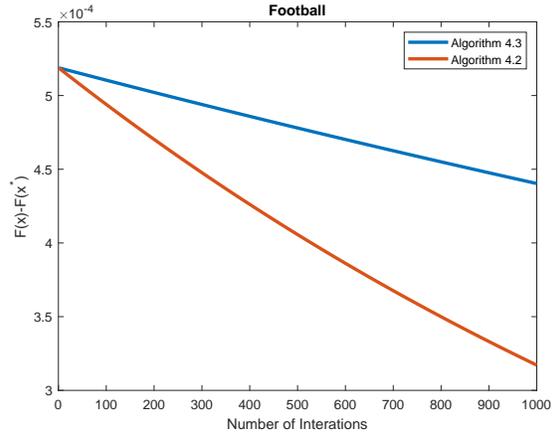
5.2. Link prediction problem. To further analyze the proposed algorithm, we depict the practical application of proposed Algorithm (4.2) to solve a link prediction problem. The Algorithm (4.2) is applied to predict missing links in networks (popularly known as link prediction [32, 33]). The link prediction is considered as the binary classification problem where the two classes are the link existence and link absence between two nodes. Logistic model [34] is used to classify the different links, which can be formulated as convex minimization problem:

$$\min_{\Theta \in \mathbb{R}^n} -\frac{1}{m} \left[\sum_{i=1}^m b^i \log h_{\Theta}(x^i) + (1 - b^i) \log(1 - h_{\Theta}(x^i)) \right] + \rho \|\Theta\|_1, \quad (5.1)$$

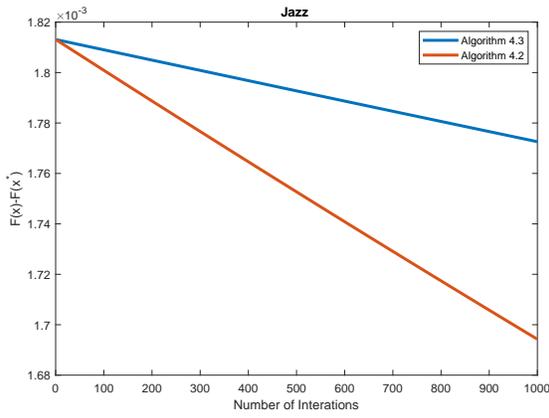
where $h_{\Theta}(u) = (1 + \exp -\Theta^T u)^{-1}$ is a sigmoid function, ρ is a regularization parameter, m is the total number of node pairs, x^i 's are feature vectors and b^i 's indicates the existence of link between nodes. Minimization problem (5.1) reduces to saddle point problem (4.1) by assuming



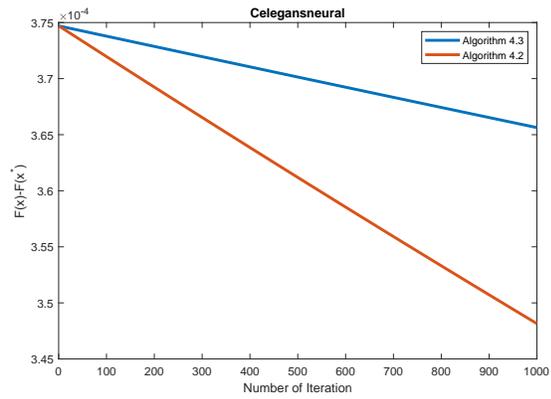
(A) Dolphin.



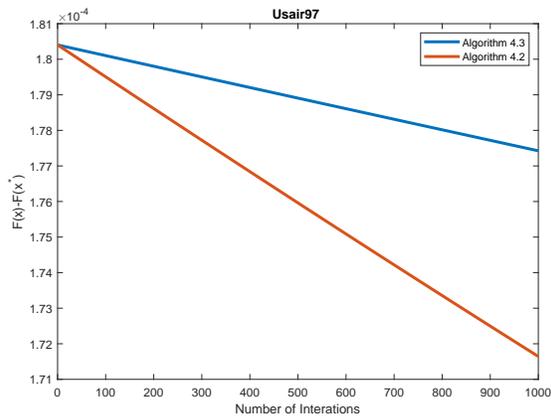
(B) Football.



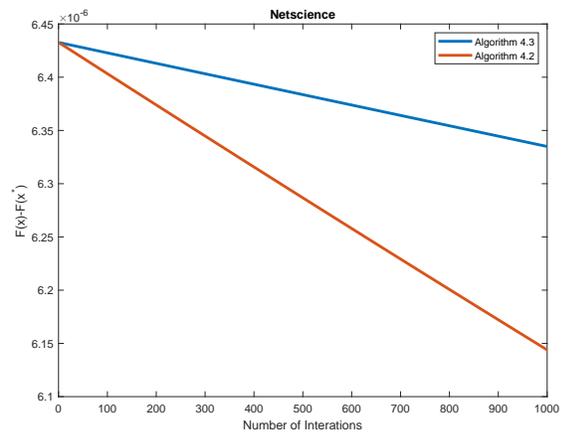
(C) Jazz.



(D) Celegansneural

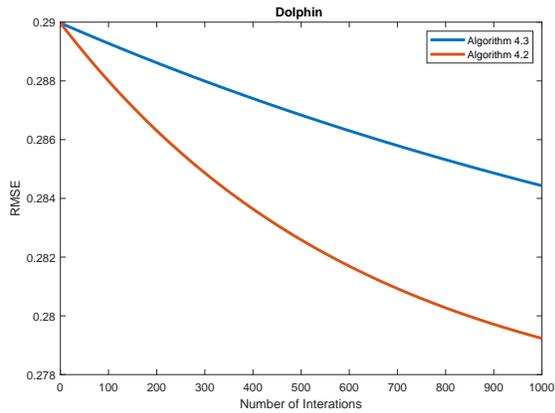


(E) Usair97

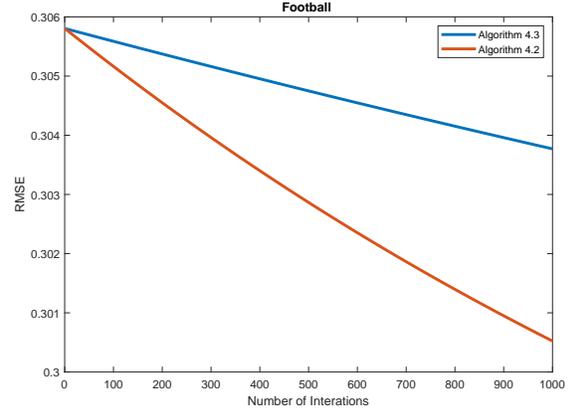


(F) Netscience.

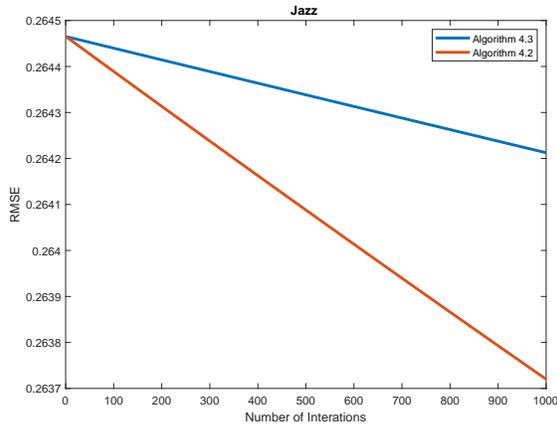
FIGURE 2. Value of $F(x_n) - F(x^*)$ for 1000 iterations with different datasets.



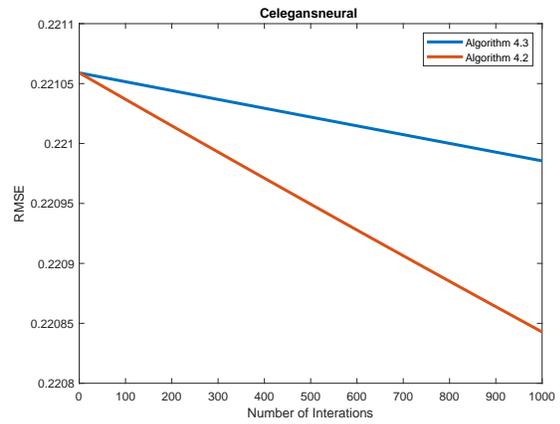
(A) Dolphin.



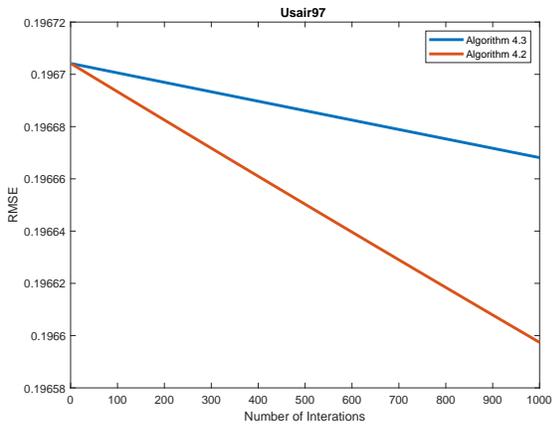
(B) Football.



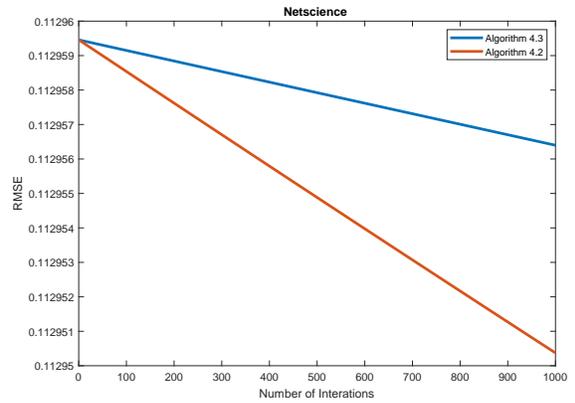
(C) Jazz.



(D) Celegansneural



(E) Usair97



(F) Netscience.

FIGURE 3. Behavior of root mean square error (RMSE) for different datasets.

$f = \rho \|\Theta\|_1$, $g = \frac{1}{m} [\sum_{i=1}^m b^i \log h_{\Theta}(x^i) + (1 - b^i) \log(1 - h_{\Theta}(x^i))]$ and $h^* = k^* = L = 0$. The experiment of the link prediction is carried out in two phases, viz., feature extraction phase and regression phase. Features are automatically extracted using the autoencoder framework of deep learning [35] with two hidden layers. Each node of the network is represented using

TABLE 3. Result

Datasets	Accuracy	Logistic Error
Dolphin	0.915918	0.084082
Football	0.906484	0.093516
Celegansneural	0.951133	0.048867
Usair97	0.961307	0.038693
Political blogs	0.734417	0.265583

TABLE 4. Result Comparison

Datasets	Accuracy					
	Algorithm (4.2)	CN	AA	RA	PA	CAR
Dolphin	0.915918	0.767566	0.759134	0.773187	0.603709	0.682928
Football	0.906484	0.658252	0.658417	0.663202	0.526189	0.846447
Jazz	0.930058	0.688999	0.698540	0.719959	0.618850	0.861920
Celegansneural	0.951133	0.539677	0.715186	0.752523	0.684004	0.842770
Usair97	0.961307	0.660190	0.782063	0.820363	0.780785	0.939876
Netscience	0.986668	0.996769	0.996824	0.996742	0.719727	0.999697
Political blogs	0.734417	0.765682	0.777344	0.853749	0.816781	0.964388

16 features. Once the node features are extracted, edge features are computed using the binary operator. Further, the regression is applied for the best estimation of the decision boundary and accuracy is computed based on this decision boundary. The experiment is carried out on real network datasets tabulated in the Table 2.

Accuracy. Accuracy and logistic error corresponding to seven real-world network datasets are shown in Table 3. The proposed method shows errors less than 10% on all datasets except the political blogs where the error reaches up to a higher level of 26.5%. It shows best accuracy result on coauthorship data (Netscience) compared to others.

We also compare the accuracy results of the proposed methods with some well known existing approaches (viz., common neighbors (CN) [33], Adamic/Adar (AA) [36], Resource allocation (RA) [37], Preferential attachment (PA) [38], and CAR [39]). These results are tabulated in Table 4, where the best value against each dataset is shown in bold-face. From Table 4, we observe that the proposed method shows best results on Dolphin, Football, Jazz, Celegansneural, and Usair97 datasets with significant margins. CAR is the best performing method on Netscience and Political blogs datasets. One thing to note that the accuracy of all the methods in the table are almost similar on Netscience except the PA.

Acknowledgments

The authors are grateful to the referees for useful suggestions which improved the contents of this paper. A. Dixit thanks IIT (BHU) for the fellowship in form of Teaching Assistantship, India. The research of J.C. Yao was partially supported by the grant MOST 108-2115-M-039-005 -MY3.

REFERENCES

- [1] P.L. Combettes, V. R. Wajs, Signal recovery by proximal forward-backward splitting, *Multiscale Model. Simul.* 4 (2005), 1168-1200.
- [2] I. Daubechies, M. Defrise, C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, *Commun. Pure Appl. Math.* 57 (2004), 1413-1457.
- [3] L.M. Briceno-Arias, P.L. Combettes, Monotone operator methods for nash equilibria in non-potential games, in: *Computational and Analytical Mathematics*, pp. 143-159, Springer, 2013.
- [4] N.T. An, N.M. Nam, X. Qin, Solving k-center problems involving sets based on optimization techniques, *J. Global Optim.* 76 (2020), 189–209.
- [5] A. Nemirovski, A. Juditsky, G. Lan, A. Shapiro, Robust stochastic approximation approach to stochastic programming, *SIAM J. Optim.* 19 (2009), 1574-1609.
- [6] X. Qin, N.T. An, Smoothing algorithms for computing the projection onto a Minkowski sum of convex sets, *Comput. Optim. Appl.* 74 (2019), 821-850.
- [7] P.L. Lions, B. Mercier, Splitting algorithms for the sum of two nonlinear operators, *SIAM J. Numer. Anal.* 16 (1979), 964-979.
- [8] J. F. Aujol, C. Dossal, Stability of over-relaxations for the forward-backward algorithm, application to FISTA, *SIAM J. Optim.* 25 (2015), 2408-2433.
- [9] X. Qin, S.Y. Cho, L. Wang, Strong convergence of an iterative algorithm involving nonlinear mappings of nonexpansive and accretive type, *Optimization*, 67 (2018), 1377-1388.
- [10] H.A. Abass, K.O. Aremu, L.O. Jolaoso, O.T. Mewomo, An inertial forward-backward splitting method for approximating solutions of certain optimization problems, *J. Nonlinear Funct. Anal.* 2020 (2020), Article ID 6.
- [11] S.Y. Cho, X. Qin, L. Wang, Strong convergence of a splitting algorithm for treating monotone operators, *Fixed Point Theory Appl.* 2014 (2014), Article ID 94.
- [12] P. Tseng, Applications of a splitting algorithm to decomposition in convex programming and variational inequalities, *SIAM J. Control Optim.* 29 (1991), 119-138.
- [13] H.H. Bauschke, P.L. Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*, Vol. 408, Springer, 2011.
- [14] B.T. Polyak, Some methods of speeding up the convergence of iteration methods, *USSR Comput. Math. Math. Phys.* 4 (1964), 1-17.
- [15] Y.E. Nesterov, A method for solving the convex programming problem with convergence rate $O(1/k^2)$, *Dokl. Akad. Nauk Sssr*, 269 (1983), 543-547.
- [16] F. Alvarez, H. Attouch, An inertial proximal method for maximal monotone operators via discretization of a nonlinear oscillator with damping, *Set-Valued Anal.* 9 (2001), 3-11.
- [17] A. Moudafi, M. Oliny, Convergence of a splitting inertial proximal method for monotone operators, *J. Comput. Appl. Math.* 155 (2003), 447-454.
- [18] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM J. Imaging Sci.* 2 (2009), 183-202.
- [19] S.Y. Cho, A monotone Bregan projection algorithm for fixed point and equilibrium problems in a reflexive Banach space, *Filomat*, 34 (2020), 1487–1497.
- [20] S.Y. Cho, A convergence theorem for generalized mixed equilibrium problems and multivalued asymptotically nonexpansive mappings, *J. Nonlinear Convex Anal.* 21 (2020), 1017-1026.
- [21] D. A. Lorenz, T. Pock, An inertial forward-backward algorithm for monotone inclusions, *J. Math. Imaging Vision*, 51 (2015), 311-325.
- [22] B.V. Limaye, *Functional Analysis Second Edition*, New Age, New Delhi, 1996.
- [23] A. Dixit, D.R. Sahu, A.K. Singh, T. Som, Application of a new accelerated algorithm to regression problems, *Soft Comput.* 24 (2019), 1-14.
- [24] F. Alvarez, Weak convergence of a relaxed and inertial hybrid projection-proximal point algorithm for maximal monotone operators in Hilbert spacem, *SIAM J. Optim.* 14 (2004), 773-782.
- [25] D. R. Sahu, J. C. Yao, M. Verma, K. K. Shukla, Convergence rate analysis of proximal gradient methods with applications to composite minimization problems, *Optimization*, 70 (2021), 75-100.

- [26] J. M. Bioucas-Dias, Bayesian wavelet-based image deconvolution: a GEM algorithm exploiting a class of heavy-tailed priors, *IEEE Trans. Image Processing* 15 (2006), 937-951.
- [27] E. J. Candes, T. Tao, Near-optimal signal recovery from random projections: Universal encoding strategies?, *IEEE Trans. Info. Theory* 52 (2006), 5406-5425.
- [28] R. Muthukrishnan, R. Rohini, LASSO: A feature selection technique in predictive modeling for machine learning. 2016 IEEE International Conference on Advances in Computer Applications (ICACA), Coimbatore, pp. 18-20, 2016. Doi: 10.1109/ICACA.2016.7887916.
- [29] A. Bruckstein, D. Donoho, M. Elad, From sparse solutions of systems of equations to sparse modeling of signals and images, *SIAM Rev.* 51 (2209), 34–81,
- [30] S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit, *SIAM J. Sci. Comput.* 20 (1998), 33–61
- [31] R. Tibshirani, Regression shrinkage and selection via the lasso, *J. Roy. Statist. Soc. Ser. B* 58 (1996), 267–288.
- [32] A. Kumar, S. S. Singh, K. Singh, B. Biswas, Level-2 node clustering coefficient based link prediction, *Appl. Intelligence*, 49 (2019), 2762-2779.
- [33] D. Liben-Nowell, J. Kleinberg, The link-prediction problem for social networks, *J. Amer. Soc. Info. Sci. Tech.* 58 (2007), 1019-1031.
- [34] M. Jordan, J. Kleinberg, B. Scholkopf, *Information Science and Statistics*, Springer, 2006.
- [35] T.N. Kipf, M. Welling, Variational Graph Auto-encoders, *CoRR* abs/1611.07308. arXiv:1611.07308, 2016.
- [36] A. Lada, E. Adar, Friends and neighbors on the web, *Social Networks* 25 (2003), 211-230.
- [37] Q. Ou, Y.D. Jin, T. Wang, B.H. Zhou, B.Q. Yin, Power-law strength-degree correlation from resource-allocation dynamics on weighted networks, *Physical Rev. E* 75 (2007), 021102. doi: 10.1103/physreve.75.021102.
- [38] A.L. Barabasi, R. Albert, Emergence of scaling in random networks, *Science* 286 (1999), 509-512.
- [39] C.V. Cannistraci, G. Alanis-Lobato, T. Ravasi, From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks, *Scientific Reports* 3 (2013), 1613. doi:10.1038/srep01613.