J. Nonlinear Var. Anal. 8 (2024), No. 3, pp. 473-484
Available online at http://jnva.biemdas.com
https://doi.org/10.23952/jnva.8.2024.3.08

A MULTI-STEP INERTIAL ASYNCHRONOUS SEQUENTIAL ALGORITHM FOR COMMON FIXED POINT PROBLEMS

FANG SU[1], LULU LIU[2], XIAO-HUAN LI[3], QIAO-LI DONG[2,*]

[1]*Department of Mathematics, National University of Defense Technology, Changsha 410073, China*
[2]*Tianjin Key Laboratory for Advanced Signal Processing and College of Science, Civil Aviation University of China, Tianjin 300300, China*
[3]*School of Mathematics and Statistics, Shandong University of Technology, Zibo 255000, China*

Abstract. In this paper, we introduce a multi-step inertial asynchronous sequential (MIAS) algorithm for common fixed point problems of nonexpansive operators and establish the weak convergence of the proposed algorithm. The unconditional convergence of a one-step inertial algorithm is also obtained. The application to linear systems is presented by combining Kaczmarz method. Finally, a numerical example for CT problems illustrates the efficiency of our algorithm.

Keywords. Asynchronous sequential inertial methods; Multi-step inertial methods; Nonexpansive operator; X-ray CT problem.

1. INTRODUCTION

Chazan and Miranker [1] first introduced the asynchronous relaxations method for the solutions of linear systems where each processing node performs its computations independently. The asynchronous methods avoid the drawbacks of the synchronous methods, and their speed is limited by the slowest node. They are robust to fail transmissions through a network and may simplify the software code to implement such algorithms. Since its inception, the asynchronous algorithm has received much attention due to its advantage in the computation. Especially, in the recent years, there are growing interests in studying asynchronous algorithms [2, 3, 4, 5, 6] since its extensive application in the deep learning and signal processing. The asynchronous algorithms for the general fixed point problems were investigated due to the relation of the fixed point problems and optimization problems [7]. Baudet [8] early applied the totally asyncparallel algorithm for a fixed point of P-contraction operators. Peng et al. [9] proposed an asynchronous algorithmic framework to find a fixed point of a nonexpansive operator where the updates are generated on random blocks of coordinates by using potentially out-of-date information. Recently, Stathopoulos and Jones [10] proposed an inertial asynchronous and parallel fixed-point iteration where several new versions of existing convex optimization algorithms were introduced.

*Corresponding author.
E-mail address: dongql@lsec.cc.ac.cn (Q.L. Dong).
Received 14 October 2023; Accepted 27 January 2024; Published online 5 April 2024.

Now we focus on the common fixed points problem. Let $\mathscr{H}$ be a Hilbert space. Recall that an operator $T:\mathscr{H}\to\mathscr{H}$ is called *nonexpansive* if $\|Tx-Ty\|\leq\|x-y\|$ for all $x,y\in\mathscr{H}$. From now on, $\mathrm{Fix}(T):=\{x\in\mathscr{H}:Tx=x\}$ is borrowed to denote the set of fixed points of T.

We give the definition of the common fixed points problem as following.

Problem 1.1. *Suppose that $\{T_i\}_{i=1}^m:\mathscr{H}\to\mathscr{H}$ are nonexpansive operators with $\bigcap_{i=1}^m\mathrm{Fix}(T_i)\neq\emptyset$. Find $x^*\in\mathscr{H}$ such that $x^*\in\bigcap_{i=1}^m\mathrm{Fix}(T_i)$.*

There are numerous numerical methods to find a common fixed points problem of a finite family of nonexpansive operators; see, e.g., [11, 12, 13, 14, 15]. Recently, by using the asynchronous idea, Heaton and Censor [16] proposed an asynchronous sequential inertial iterations (ASI) for Problem 1.1.

Suppose that there is a collection of w processing nodes. For each iteration index k, let $d_k\in\mathbb{Z}^w_{\geq 0}$ give the delay information for each of the nodes. This means that if the last output from the i-th node N_i was computed by using the iterate x^{k-j}, then the i-th component of d^k gives this delay amount, i.e., $(d^k)_i=j$. Let $\{i_k\}_{k\in\mathbb{N}}$ be an index sequence identifying the index of the operator whose output is used to compute x^{k+1}, where $i_k\in\{1,2,\ldots,m\}$ for all $k\in\mathbb{N}$. Then $\hat{x}^k$ is the last iterate sent to the i_k-th node, and we write

$$\hat{x}^k:=x^{k-(d^k)_{i_k}},\quad \text{for all } k\in\mathbb{N}.$$

Let $[m]:=\{1,2,\ldots,m\}$ and define

$$S_i:=\mathrm{Id}-T_i,\quad \text{for all } i\in[m], \tag{1.1}$$

where Id is the identity operator.

The algorithm of Heaton and Censor [16] is as follows.

Algorithm 1 (Asynchronous Sequential Inertial (ASI) Algorithm)

Let $x^1\in\mathscr{H}$ be arbitrary, $\{\lambda_k\}_{k\in\mathbb{N}}$ be such that $\lambda_k\in(0,1)$ for all $k\in\mathbb{N}$, and $\{i_k\}_{k\in\mathbb{N}}$ be an almost cyclic control on $[m]$[1]. For each $k\in\mathbb{N}$, set

$$x^{k+1}=\begin{cases}x^k, & \text{if } k\leq\sup_{k\in\mathbb{N}}\|d^k\|_\infty,\\ x^k-\lambda_k S_{i_k}(\hat{x}^k), & \text{otherwise.}\end{cases} \tag{1.2}$$

By using (1.1), for $k>\sup_{k\in\mathbb{N}}\|d^k\|_\infty$, the expression of x^{k+1} in (1.2) can be rewritten as follows:

$$\begin{aligned}x^{k+1}&=(1-\lambda_k)x^k+\lambda_k(x^k-\hat{x}^k)+\lambda_k T_{i_k}(\hat{x}^k)\\&=(1-\lambda_k)\left(x^k+\frac{\lambda_k}{1-\lambda_k}(x^k-\hat{x}^k)\right)+\lambda_k T_{i_k}(\hat{x}^k).\end{aligned}$$

Since

$$x^k-\hat{x}^k=\sum_{j=0}^{(d^k)_{i_k}-1}\left(x^{k-j}-x^{k-j-1}\right),$$

Algorithm 1 is a special case of the multi-step inertial Krasnosel'skiĭ-Mann iterations [17].

[1] See Definition 2.3

2. Preliminaries

We use the following notations:

- $\rightharpoonup$ for the weak convergence and $\to$ for the strong convergence;
- $\omega_w(x^k) = \{x : \exists x^{k_j} \rightharpoonup x\}$ denotes the weak ω-limit set of $\{x^k\}_{k\in\mathbb{N}}$.

Definition 2.1. Let $K \subseteq \mathscr{H}$ be nonempty. An operator $T : K \to \mathscr{H}$ is said to be α-inverse strongly monotone if there exists a constant $\alpha > 0$ such that $\langle T(x) - T(y), x - y\rangle \geq \alpha\|T(x) - T(y)\|^2$ for all $x, y \in K$.

Definition 2.2. Let $T : \mathscr{H} \to \mathscr{H}$ be an operator and $\alpha \in [0,1]$. The operator $T_\alpha : \mathscr{H} \to \mathscr{H}$ defined by $T_\alpha := (1-\alpha)\mathrm{Id} + \alpha T$ is called an α-relaxation of the operator T.

It is easy to verify that T_α is nonexpansive when T is nonexpansive.

Definition 2.3 ([16])**.** A sequence $\{i_k\}_{k\in\mathbb{N}}$ is called an almost cyclic control on $[m] := \{1,2,...,m\}$ if $i_k \in [m]$ for all $k \in \mathbb{N}$ and there exists an integer $M \geq m$ (called the almost cyclicality constant) such that, for each $k \in \mathbb{N}$, $[m] \subseteq \{i_{k+1}, i_{k+2}, ..., i_{k+M}\}$.

Lemma 2.1 ([16])**.** *Let $\{T_i\}_{i=1}^m$ be a finite family of nonexpansive operators with a common fixed point, and let $y \in \mathscr{H}$ be a weak cluster point of a sequence $\{x^k\}_{k\in\mathbb{N}}$. If $\|T_i x^k - x^k\| \to 0$ for all $i \in [m]$, then $y \in \bigcap_{i=1}^m \mathrm{Fix}(T_i)$.*

Lemma 2.2 ([16])**.** *Let $\{T_i\}_{i=1}^m$ be a family of nonexpansive operators on $\mathscr{H}$ with a common fixed point, and let $\{x^k\}_{k\in\mathbb{N}}$ be a sequence in $\mathscr{H}$. If, for every $z \in \bigcap_{i=1}^m \mathrm{Fix}(T_i)$, the sequence $\{\|x^k - z\|\}_{k\in\mathbb{N}}$ converges and $\|T_i x^k - x^k\| \to 0$ for all $i \in [m]$, then $\{x^k\}_{k\in\mathbb{N}}$ converges weakly to some point in $\bigcap_{i=1}^m \mathrm{Fix}(T_i)$.*

Using the Cauchy-Schwarz and the mean value inequalities, it is easy to verify the following lemma.

Lemma 2.3. *For any $a, b \in \mathscr{H}$, the following holds: $\|a-b\|^2 \leq (1+\|b\|)\|a\|^2 + \|b\| + \|b\|^2$.*

Lemma 2.4 ([18])**.** *Let $\{a_k\}_{k\in\mathbb{N}}$ and $\{\varpi_k\}_{k\in\mathbb{N}}$ be nonnegative sequences, and let $\sum_{k=1}^\infty \varpi_k < +\infty$, $\sum_{k=1}^\infty \beta_k < +\infty$, and $a_{k+1} \leq (1+\varpi_k)a_k + \beta_k$, then $\lim_{k\to\infty} a_k$ exists.*

3. The Multi-step Inertial Asynchronous Sequential (MIAS) Algorithm

To solve Problem 1.1, we propose a multi-step inertial asynchronous sequential (MIAS) algorithm as follows:

Algorithm 2 (Multi-step Inertial Asynchronous Sequential (MIAS) Algorithm)

Step 1 Let $x^{-1}, x^0 \in \mathscr{H}$ be arbitrary, and let $\{\lambda_k\}_{k\in\mathbb{N}}$ be such that $\lambda_k \in (0,1)$ for all $k \in \mathbb{N}$. Let $\{i_k\}_{k\in\mathbb{N}}$ be an almost cyclic control on $[m]$.

Step 2 For each $k \in \mathbb{N}$, let $I(k) \subseteq \{0,1,2,\cdots,k-1\}$ and $J(k) \subseteq \{k-1,\ldots,k-\tau\}$, where τ is a finite maximal delay. Calculate

$$\begin{cases} y^k = x^k - \sum_{d\in I(k)} \alpha_{d,k}(x^d - x^{d+1}), \\ \hat{x}^k = x^k + \sum_{d\in J(k)} (x^d - x^{d+1}), \\ x^{k+1} = y^k - \lambda_k S_{i_k}\hat{x}^k. \end{cases} \tag{3.1}$$

Now we provide some explanations for the MIAS algorithm. From (1.1) and (3.1), it follows

$$\begin{aligned} x^{k+1} &= y^k - \lambda_k(\hat{x}^k - T_{i_k}(\hat{x}^k)) \\ &= (1-\lambda_k)y^k + \lambda_k T_{i_k}(\hat{x}^k) + \lambda_k(y^k - \hat{x}^k) \\ &= z^k + \lambda_k(y^k - \hat{x}^k), \end{aligned}$$

where $z^k = (1-\lambda_k)y^k + \lambda_k T_{i_k}(\hat{x}^k)$. It is obvious that the computation of x^{k+1} can be expressed in two parts. The first part is convex combination of y^k and $T_{i_k}(\hat{x}^k)$ to form the point z^k. The second part is an inertial term, given y^k and $\hat{x}^k$, that estimates the direction of the solution.

To illustrate the MIAS algorithm graphically, let C_1 and C_2 be two closed and convex sets with nonempty intersection, and let T_1 and T_2 be relaxations of the projections P_{C_1} and P_{C_2} onto sets C_1 and C_2, respectively. In this case, Figure 1 was modified based on the one from [16] which shows how x^{k+1} is generated from y^k and $\hat{x}^k$.

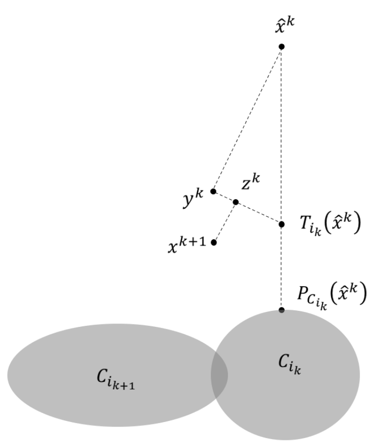

FIGURE 1. Illustration of a full iteration of the MIAS algorithm with two convex sets and the T_i's as relaxed projections onto the sets.

Remark 3.1. We review the relations between the MIAS algorithm and other methods.

(i) Let $I(k) = \{k-1\}$ and $\alpha_{k-1,k} := \alpha_k$, $\forall k \in \mathbb{N}$. The MIAS algorithm reduces to the one-step inertial asynchronous sequential (OIAS) iteration:

$$\begin{cases} y^k = x^k + \alpha_k(x^k - x^{k-1}), \\ \hat{x}^k = x^k + \sum_{d\in J(k)} (x^d - x^{d+1}), \\ x^{k+1} = y^k - \lambda_k S_{i_k}\hat{x}^k. \end{cases} \tag{3.2}$$

(ii) Let $I(k)=\emptyset$ for all $k\in\mathbb{N}$. The OIAS algorithm (3.2) reduces to Algorithm 1. Further, let $J(k)=\emptyset$ for all $k\in\mathbb{N}$. Then, we obtain the following sequential iteration: $x^{k+1}=x^k-\lambda_k S_{i_k}x^k$, which is the classical Krasnosel'skiĭ-Mann iteration.

(iii) The OIAS algorithm (3.2) is a special case of the multi-step Krasnosel'skiĭ-Mann (MiKM) iteration in [17]. Indeed, from the definition of y^k and $\hat{x}^k$, we can obtain

$$\begin{aligned} x^{k+1}=&y^k-\lambda_k\hat{x}^k+\lambda_k T_{i_k}(\hat{x}^k)\\ =&x^k-\sum_{d\in I(k)}\alpha_{d,k}(x^d-x^{d+1})-\lambda_k x^k-\lambda_k\sum_{d\in J(k)}(x^d-x^{d+1})+\lambda_k T_{i_k}(\hat{x}^k)\\ =&(1-\lambda_k)\big(x^k-\frac{1}{1-\lambda_k}\sum_{d\in I(k)}\alpha_{d,k}(x^d-x^{d+1})-\frac{\lambda_k}{1-\lambda_k}\sum_{d\in J(k)}(x^d-x^{d+1})\big)\\ &+\lambda_k T_{i_k}(\hat{x}^k)\\ =&(1-\lambda_k)w^k+\lambda_k T_{i_k}(\hat{x}^k), \end{aligned} \tag{3.3}$$

where

$$w^k=x^k-\frac{1}{1-\lambda_k}\sum_{d\in I(k)}\alpha_{d,k}(x^d-x^{d+1})-\frac{\lambda_k}{1-\lambda_k}\sum_{d\in J(k)}(x^d-x^{d+1}).$$

It is easy to verify that (3.3) is a special case of the MiKM iteration. Therefore the convergence guarantees of the MiKM iteration cover OIAS provided that the parameters $\{\alpha_{d,k}\}_{k\in\mathbb{N},d\in I(k)}$ and $\{\lambda_k\}_{k\in\mathbb{N}}$ satisfy some conditions.

4. Convergence Analysis

To establish the convergence of the MIAS algorithm, we make the following assumption:

$$\sum_{k=0}^{\infty}\Big\|\sum_{d\in I(k)}\alpha_{d,k}(x^d-x^{d+1})\Big\|<\infty. \tag{4.1}$$

Remark 4.1. If the inertial parameter sequence $\{\alpha_{d,k}\}_{k\in\mathbb{N},d\in I(k)}$ is chosen in $[0,1]$, then condition (4.1) can be simplified to

$$\sum_{k=0}^{\infty}\max_{d\in I(k)}\{\alpha_{d,k}\}\sum_{d\in I(k)}\|x^d-x^{d+1}\|<\infty. \tag{4.2}$$

Condition (4.2) can be enforced by a simple online updating rule, such as, for each $d\in I(k)$ and $\alpha_k\in[0,1]$, $\alpha_{d,k}=\min\{\alpha_k,c_{d,k}\}$, where $c_{d,k}>0$ and $c_{d,k}\sum_{d\in I(k)}\|x^d-x^{d+1}\|<+\infty$. For instance, one can choose

$$c_{d,k}=\frac{f_{d,k}}{k^{1+\delta}\sum_{d\in I(k)}\|x^d-x^{d+1}\|},\ \ f_{d,k}>c>0,\ \ \delta>0.$$

In practice, most of the time, with the proper choice of each α_k, $\alpha_{d,k}=\min\{\alpha_k,c_{d,k}\}$ may never be triggered.

Lemma 4.1. *Let $z \in \bigcap_{i=1}^{m} \mathrm{Fix}(T_i)$. Let $\{x^k\}_{k\in\mathbb{N}}$ be any sequence generated by the MIAS algorithm. If the delay vectors are uniformly bounded in the sup norm by some $\tau \geq 0$, then*

$$\begin{aligned}\|x^{k+1}-z\|^2 \leq &\Big(1+\Big\|\sum_{d\in I(k)}\alpha_{d,k}(x^d-x^{d+1})\Big\|\Big)\|x^k-z\|^2+\mu\sum_{d\in J(k)}\|x^d-x^{d+1}\|^2\\ &+\lambda_k\Big[\frac{\lambda_k}{\mu}(1+\tau)+\lambda_k-1\Big]\|S_{i_k}\hat{x}^k\|^2+(1+\mu)\Big\|\sum_{d\in I(k)}\alpha_{d,k}(x^d-x^{d+1})\Big\|^2\\ &+\Big\|\sum_{d\in I(k)}\alpha_{d,k}(x^d-x^{d+1})\Big\|,\end{aligned} \tag{4.3}$$

where μ is a positive constant.

Proof. From the definition of x^{k+1}, we have

$$\begin{aligned}\|x^{k+1}-z\|^2 =&\|y^k-z\|^2-2\lambda_k\langle S_{i_k}\hat{x}^k, y^k-z\rangle+\lambda_k^2\|S_{i_k}\hat{x}^k\|^2\\ =&\|y^k-z\|^2-2\lambda_k\langle S_{i_k}\hat{x}^k, y^k-\hat{x}^k\rangle-2\lambda_k\langle S_{i_k}\hat{x}^k,\hat{x}^k-z\rangle+\lambda_k^2\|S_{i_k}\hat{x}^k\|^2.\end{aligned} \tag{4.4}$$

By using Lemma 2.3 and the definition of y^k, we have

$$\begin{aligned}\|y^k-z\|^2 =&\Big\|x^k-\sum_{d\in I(k)}\alpha_{d,k}(x^d-x^{d+1})-z\Big\|^2\\ \leq&\Big(1+\Big\|\sum_{d\in I(k)}\alpha_{d,k}(x^d-x^{d+1})\Big\|\Big)\|x^k-z\|^2+\Big\|\sum_{d\in I(k)}\alpha_{d,k}(x^d-x^{d+1})\Big\|\\ &+\Big\|\sum_{d\in I(k)}\alpha_{d,k}(x^d-x^{d+1})\Big\|^2.\end{aligned} \tag{4.5}$$

Observe that

$$\begin{aligned}&-2\lambda_k\langle S_{i_k}\hat{x}^k, y^k-\hat{x}^k\rangle\\ =&2\lambda_k\Big\langle S_{i_k}\hat{x}^k, \sum_{d\in I(k)}\alpha_{d,k}(x^d-x^{d+1})+\sum_{d\in J(k)}(x^d-x^{d+1})\Big\rangle\\ =&2\lambda_k\Big\langle S_{i_k}\hat{x}^k, \sum_{d\in I(k)}\alpha_{d,k}(x^d-x^{d+1})\Big\rangle+2\lambda_k\sum_{d\in J(k)}\Big\langle S_{i_k}\hat{x}^k, x^d-x^{d+1}\Big\rangle\\ \leq&2\lambda_k\Big(\|S_{i_k}\hat{x}^k\|\Big\|\sum_{d\in I(k)}\alpha_{d,k}(x^d-x^{d+1})\Big\|+\sum_{d\in J(k)}\|S_{i_k}\hat{x}^k\|\|x^d-x^{d+1}\|\Big)\\ \leq&\lambda_k\Big(\frac{\lambda_k}{\mu}\|S_{i_k}\hat{x}^k\|^2+\frac{\mu}{\lambda_k}\Big\|\sum_{d\in I(k)}\alpha_{d,k}(x^d-x^{d+1})\Big\|^2+\frac{\lambda_k\tau}{\mu}\|S_{i_k}\hat{x}^k\|^2\\ &+\frac{\mu}{\lambda_k}\sum_{d\in J(k)}\|x^d-x^{d+1}\|^2\Big)\\ =&\frac{\lambda_k^2}{\mu}(1+\tau)\|S_{i_k}\hat{x}^k\|^2+\mu\Big(\Big\|\sum_{d\in I(k)}\alpha_{d,k}(x^d-x^{d+1})\Big\|^2+\sum_{d\in J(k)}\|x^d-x^{d+1}\|^2\Big).\end{aligned} \tag{4.6}$$

Since S_{i_k} is $\frac{1}{2}$-cocoercive and $S_{i_k}z = 0$, we have

$$\|S_{i_k}\hat{x}^k\|^2 = \|S_{i_k}\hat{x}^k - S_{i_k}z\|^2 \leq 2\langle S_{i_k}\hat{x}^k - S_{i_k}z, \hat{x}^k - z\rangle = 2\langle S_{i_k}\hat{x}^k, \hat{x}^k - z\rangle,$$

which implies $-2\lambda_k\langle S_{i_k}\hat{x}^k, \hat{x}^k - z\rangle \leq -\lambda_k\|S_{i_k}\hat{x}^k\|^2$, which together with (4.4)-(4.6) yields (4.3). This completes the proof. ☐

Lemma 4.2. *Let $z \in \bigcap_{i=1}^m \mathrm{Fix}(T_i)$. Let $\{x^k\}_{k\in\mathbb{N}}$ be the sequence generated by the MIAS algorithm and*

$$\xi_k := \|x^k - z\|^2 + \sum_{d=k-\tau}^{k-1} c_{d,k}\|x^d - x^{d+1}\|^2, \tag{4.7}$$

where $c_{d,k} = \big[d-(k-\tau)+2\big]\mu$, and μ is a positive constant. Assume that the delay vectors are uniformly bounded in the sup norm by some $\tau \geq 0$ and that (4.1) holds. If $\{\lambda_k\}_{k\in\mathbb{N}}$ satisfies

$$0 < \lambda_k \leq \frac{1}{1+2\sqrt{2}(\tau+1)}, \quad \forall k \in \mathbb{N}, \tag{4.8}$$

then

(i) *the sequence $\{\xi_k\}_{k\in\mathbb{N}}$ is a convergent sequence;*
(ii) *the sequence $\{\|x^k - z\|\}_{k\in\mathbb{N}}$ converges.*

Proof. (i): From (3.1), it follows

$$\|x^{k+1} - x^k\|^2 \leq 2\lambda_k^2\|S_{i_k}\hat{x}^k\|^2 + 2\Big\|\sum_{d\in I(k)} \alpha_{d,k}(x^d - x^{d+1})\Big\|^2. \tag{4.9}$$

From the definition of ξ_k and (4.3), we obtain

$$\begin{aligned}
\xi_{k+1} =& \|x^{k+1} - z\|^2 + \sum_{d=k+1-\tau}^{k} c_{d,k+1}\|x^d - x^{d+1}\|^2 \\
\leq& \Big(1 + \Big\|\sum_{d\in I(k)} \alpha_{d,k}(x^d - x^{d+1})\Big\|\Big)\xi_k + \lambda_k\Big[\frac{\lambda_k}{\mu}(1+\tau) + \lambda_k - 1\Big]\|S_{i_k}\hat{x}^k\|^2 \\
&+ (1+\mu)\Big\|\sum_{d\in I(k)} \alpha_{d,k}(x^d - x^{d+1})\Big\|^2 + \Big\|\sum_{d\in I(k)} \alpha_{d,k}(x^d - x^{d+1})\Big\| \\
&+ \sum_{d=k+1-\tau}^{k} c_{d,k+1}\|x^d - x^{d+1}\|^2 + \mu\sum_{d=k-\tau}^{k-1}\|x^d - x^{d+1}\|^2 \\
&- \Big\|\sum_{d\in I(k)} \alpha_{d,k}(x^d - x^{d+1})\Big\|\sum_{d=k-\tau}^{k-1} c_{d,k}\|x^d - x^{d+1}\|^2 - \sum_{d=k-\tau}^{k-1} c_{d,k}\|x^d - x^{d+1}\|^2 \\
\leq& \Big(1 + \Big\|\sum_{d\in I(k)} \alpha_{d,k}(x^d - x^{d+1})\Big\|\Big)\xi_k + \lambda_k\Big[\frac{\lambda_k}{\mu}(1+\tau) + \lambda_k - 1\Big]\|S_{i_k}\hat{x}^k\|^2 \\
&+ (1+\mu)\Big\|\sum_{d\in I(k)} \alpha_{d,k}(x^d - x^{d+1})\Big\|^2 + \Big\|\sum_{d\in I(k)} \alpha_{d,k}(x^d - x^{d+1})\Big\| \\
&+ \mu(\tau+1)\|x^k - x^{k+1}\|^2 - \mu\|x^{k-\tau} - x^{k-\tau+1}\|^2.
\end{aligned}$$

Using (4.8) and (4.9), we have

$$\begin{aligned}\xi_{k+1} \leq & \left(1+\Big\|\sum_{d\in I(k)}\alpha_{d,k}(x^d-x^{d+1})\Big\|\right)\xi_k+\big[1+(2\tau+3)\mu\big]\Big\|\sum_{d\in I(k)}\alpha_{d,k}(x^d-x^{d+1})\Big\|^2 \\ & +\Big\|\sum_{d\in I(k)}\alpha_{d,k}(x^d-x^{d+1})\Big\|-\mu\|x^{k-\tau}-x^{k-\tau+1}\|^2 \\ & +\lambda_k\left[\frac{\lambda_k}{\mu}+\frac{\lambda_k\tau}{\mu}+\lambda_k-1+2\lambda_k(\tau+1)\mu\right]\|S_{i_k}\hat{x}^k\|^2.\end{aligned} \tag{4.10}$$

In order to see $\lambda_k\left[\frac{\lambda_k}{\mu}+\frac{\lambda_k\tau}{\mu}+\lambda_k-1+2\lambda_k(\tau+1)\mu\right]<0$, we let $\lambda>0$ and

$$\frac{\lambda_k}{\mu}\left[1+\tau+\mu+2(\tau+1)\mu^2\right]<1.$$

Then,

$$0<\lambda_k\leq\frac{\mu}{1+\tau+\mu+2(\tau+1)\mu^2}. \tag{4.11}$$

Since $\frac{\mu}{1+\tau+\mu+2(\tau+1)\mu^2}$ reaches the maximum $\frac{1}{1+2\sqrt{2}(\tau+1)}$ if one takes $\mu=\frac{\sqrt{2}}{2}$, so $\mu=\frac{\sqrt{2}}{2}$ is an optimal parameter of (4.11). In this case, (4.11) becomes

$$0<\lambda_k\leq\frac{1}{1+2\sqrt{2}(\tau+1)}.$$

Thus it follows from (4.10) that

$$\begin{aligned}\xi_{k+1} \leq & \left(1+\Big\|\sum_{d\in I(k)}\alpha_{d,k}(x^d-x^{d+1})\Big\|\right)\xi_k+\left[1+\frac{\sqrt{2}}{2}+\sqrt{2}(\tau+1)\right]\Big\|\sum_{d\in I(k)}\alpha_{d,k}(x^d-x^{d+1})\Big\|^2 \\ & +\Big\|\sum_{d\in I(k)}\alpha_{d,k}(x^d-x^{d+1})\Big\|-\frac{\sqrt{2}}{2}\|x^{k-\tau}-x^{k-\tau+1}\|^2.\end{aligned} \tag{4.12}$$

Using Lemma 2.4, (4.1), and (4.12), we obtain that $\lim_{k\to\infty}\xi_k$ exists.

(ii): From (4.12), we have

$$\begin{aligned}\mu\|x^{k-\tau}-x^{k-\tau+1}\|^2 \leq & \xi_k-\xi_{k+1}+(\xi_k+1)\Big\|\sum_{d\in I(k)}\alpha_{d,k}(x^d-x^{d+1})\Big\| \\ & +\big[1+\mu+2(\tau+1)\mu\big]\Big\|\sum_{d\in I(k)}\alpha_{d,k}(x^d-x^{d+1})\Big\|^2.\end{aligned}$$

Since $\lim_{k\to\infty}\xi_k$ exists, we obtain by using (4.1) that $\lim_{k\to\infty}\|x^k-x^{k+1}\|=0$, which together with (4.7) obtains the existence of $\lim_{k\to\infty}\|x^k-z\|$. This completes the proof. □

Lemma 4.3. *Let $\{x^k\}_{k\in\mathbb{N}}$ be a sequence generated by the MIAS algorithm. Assume that (4.1) is satisfied and $\lambda_k\geq\varepsilon$, where $\varepsilon>0$ is a given constant. If the delay vectors are uniformly bounded in the sup norm by some τ, then $\|T_ix^k-x^k\|\to 0$, for each $i\in[m]$.*

Proof. Let $i\in[m]$ and let $T_{i,\lambda}$ be the λ-relaxation of T_i. If $\{t_k\}_{k\in\mathbb{N}}$ is an increasing sequence such that $t_k\in\mathbb{N}$ for all $k\in\mathbb{N}$, then

$$\|T_{i,\lambda_{t_k}}(x^k)-x^k\|=\lambda_{t_k}\|T_i(x^k)-x^k\|\geq\varepsilon\|T_i(x^k)-x^k\|.$$

Thus we only need to prove $\|T_{i,\lambda_{t_k}}(x^k)-x^k\|\to 0$ in order to show $\|T_i(x^k)-x^k\|\to 0$. For each $k\in\mathbb{N}$, let t_k be the smallest index greater than or equal to k such that $i_{t_k}=i$. Then

$$x^{t_k+1}=y^{t_k}-\lambda_{t_k}S_i(\hat{x}^{t_k})=y^{t_k}-\lambda_{t_k}\hat{x}^{t_k}+\lambda_{t_k}T_i\hat{x}^{t_k}=T_{i,\lambda_{t_k}}(\hat{x}^{t_k})+(y^{t_k}-\hat{x}^{t_k}).$$

It follows that

$$\begin{aligned}\|T_{i,\lambda_{t_k}}(x^k)-x^{t_k+1}\| &= \|T_{i,\lambda_{t_k}}(x^k)-T_{i,\lambda_{t_k}}(\hat{x}^{t_k})-(y^{t_k}-\hat{x}^{t_k})\|\\ &\le \|T_{i,\lambda_{t_k}}(x^k)-T_{i,\lambda_{t_k}}(\hat{x}^{t_k})\|+\|y^{t_k}-\hat{x}^{t_k}\|\\ &\le \|x^k-x^{t_k}\|+2\|x^{t_k}-\hat{x}^{t_k}\|+\Big\|\sum_{d\in I(t_k)}\alpha_{t_k,d}(x^d-x^{d+1})\Big\|,\end{aligned}\tag{4.13}$$

where the last inequality origins from the nonexpansiveness of $T_{i,\lambda_{t_k}}$. Due to the bound on delays and the almost cyclicity of $\{i_k\}_{k\in\mathbb{N}}$, we know that $\hat{x}^{t_k}=x^j$ for some $k-\tau\le j\le t_k\le k+M$, where M is the almost cyclicality constant of $\{i_k\}_{k\in\mathbb{N}}$. By (4.13) and (3.1), we see that

$$\begin{aligned}&\|T_{i,\lambda_{t_k}}(x^k)-x^k\|\\ \le&\|T_{i,\lambda_{t_k}}(x^k)-x^{t_k+1}\|+\|x^{t_k+1}-x^k\|\\ \le&\|x^k-x^{t_k}\|+2\|\sum_{d\in J(t_k)}(x^d-x^{d+1})\|+\Big\|\sum_{d\in I(t_k)}\alpha_{t_k,d}(x^d-x^{d+1})\Big\|+\|x^{t_k+1}-x^k\|\\ \le&\sum_{d=k}^{k+M-1}\|x^d-x^{d+1}\|+2\sum_{d=k-\tau}^{k+M-1}\|x^d-x^{d+1}\|+\sum_{l=k-\tau}^{k+M}\Big\|\sum_{d\in I(l)}\alpha_{l,d}(x^d-x^{d+1})\Big\|+\sum_{d=k}^{k+M}\|x^{d+1}-x^d\|.\end{aligned}$$

Therefore, we obtain from (4.1) that $\lim_{k\to\infty}\|T_{i,\lambda_k}(x^k)-x^k\|=0$. This completes the proof. □

Now we can state and prove the main result of this paper.

Theorem 4.1. *Let $\{x^k\}_{k\in\mathbb{N}}$ be a sequence generated by the MIAS algorithm. Assume that (4.1) is satisfied. If the delay vectors are uniformly bounded in the sup norm by some $\tau\ge 0$ and*

$$0<\varepsilon\le\lambda_k\le\frac{1}{1+2\sqrt{2}(\tau+1)},\quad \forall k\in\mathbb{N},$$

then $\{x^k\}_{k\in\mathbb{N}}$ converges weakly to a common fixed point of the family $\{T_i\}_{i=1}^m$.

Proof. Obviously, we see from Lemma 4.2 (ii), Lemma 4.3, and Lemma 2.2 that $\{x^k\}_{k\in\mathbb{N}}$ converges weakly to a common fixed point of the family $\{T_i\}_{i=1}^m$. □

From Theorem 4.1, we obtain the convergence of the OIAS algorithm (3.2).

Theorem 4.2 (Conditional Convergence for $I(k)=\{k-1\}$). *Let $\{x^k\}_{k\in\mathbb{N}}$ be a sequence generated by the OIAS algorithm. Assume $\left\|\sum_{k=1}^{\infty}\alpha_k(x^k-x^{k-1})\right\|<\infty$. If the delay vectors are uniformly bounded in the sup norm by some $\tau\ge 0$ and*

$$0<\varepsilon\le\lambda_k\le\frac{1}{1+2\sqrt{2}(\tau+1)},\quad \forall k\in\mathbb{N},$$

then the sequence $\{x^k\}_{k\in\mathbb{N}}$ converges weakly to a common fixed point of the family $\{T_i\}_{i=1}^m$.

5. Application to linear systems and numerical examples

We consider the linear systems: find $x \in \mathbb{R}^N$ such that $Ax = b$, where $A \in \mathbb{R}^{M\times N}$ and $b \in \mathbb{R}^M$. Each equation in the system can be associated with a closed and convex subset of $\mathbb{R}^N$, namely, the hyperplane $H_i = \{x \in \mathbb{R}^N \mid \langle a^i, x\rangle = b_i\}$ for all $i \in [M]$, where a^i is the i-th row of A and b_i is the i-th component of b. The projection P_i onto H_i is as follows $P_i(x) = x + \frac{b_i - \langle a^i, x\rangle}{\|a^i\|^2} a^i$. Because $2P_i - \mathrm{Id}$ is nonexpansive (see [11]), we set

$$S_i(x) = x - (2P_i - \mathrm{Id})(x) = 2\frac{\langle a^i, x\rangle - b_i}{\|a^i\|^2} a^i.$$

Then, similar to (3.1), we give the MIAS-ART method as follows:

Algorithm 3 (MIAS-ART)

Step 1 Let $A \in \mathbb{R}^{M\times N}$ and $b \in \mathbb{R}^M$ be given. Let $x^{-1}, x^0 \in \mathbb{R}^N$ be arbitrary, $\{\lambda_k\}_{k\in\mathbb{N}}$ be such that $\lambda_k \in (0,1)$ for all $k \in \mathbb{N}$, and $\{i_k\}_{k\in\mathbb{N}}$ be an almost cyclic control on $[M]$.

Step 2 For each $k \in \mathbb{N}$, let $I(k) \subseteq \{0,1,2,\cdots,k-1\}$ and $J(k) \subseteq \{k-1,\ldots,k-\tau\}$, where τ is a finite maximal delay. Calculate

$$\begin{cases} y^k = x^k - \sum_{d\in I(k)} \alpha_{d,k}(x^d - x^{d+1}), \\ \hat{x}^k = x^k + \sum_{d\in J(k)} (x^d - x^{d+1}), \\ x^{k+1} = y^k - 2\lambda_k \frac{\langle a^{i_k}, \hat{x}^k\rangle - b_{i_k}}{\|a^{i_k}\|^2} a^{i_k}. \end{cases}$$

Next, we provide a preliminariy experiment and compare our Algorithm 3 with Algorithm 3 in [16] (ASI-ART). For Algorithm 3, we only consider one step inertial algorithm (MIAS-ART I) and two-step inertial algorithm (MIAS-ART II), that is, $I(k) = \{k-1\}$ and $I(k) = \{k-1, k-2\}$. All codes were written in MATLAB R2018a and performed on a notebook with AMD Ryzen 7 5800H, RAM 16.00 GB.

We solve the X-ray CT problem generated by a MATLAB package AIR Tools II [19]. The X-rays are parallel and the distance between the first and the last ray is 50. The source-detector pair is rotated around the object, and measurements are recorded for angles from 0 to 179. Meanwhile, we reshape the exact solution into a 50×50 image. In this case, we get a 3000×2500 matrix.

The initial point $x^{-2} = x^{-1} = x^0 = 0$. In the numerical experiment, we take

$$E_k = \frac{\|x^k - x^*\|^2}{\|x^*\|^2} < 10^{-6}$$

as the stopping criterion, where x^* is the true image vector. All parameters in the algorithm are selected by manual tuning.

The results are reported in Table 1 and Figure 2. It can be seen from Table 1 that when τ is 3, MIAS-ART II is the best in iteration steps and time, and when τ is 6 and 10, the performance of MIAS-ART II and MIAS-ART I is better than that of ASI-ART in iteration steps and time.

TABLE 1. Comparison results of three algorithms under different delays.

τ		ASI-ART	MIAS-ART I	MIAS-ART II
3	Iter.	73281376	30457047	27342047
	CPU time	5612.0002	2460.8659	2359.1832
6	Iter.	129435557	32809178	32216887
	CPU time	9714.1115	2630.3745	2697.1906
10	Iter.	209377964	42422534	41473102
	CPU time	15308.2862	3624.2697	3707.3042

(a) Original CT image

(b) CT image reconstructed by the ASI-ART

(c) CT image reconstructed by the MIAS-ART I

(d) CT image reconstructed by the MIAS-ART II

FIGURE 2. CT image reconstruction when $\tau = 10$.

REFERENCES

[1] D. Chazan, W. Miranker, Chaotic relaxation, Linear Algebra Appl. 2 (1969), 199-222.

[2] D. Bertsekas, J. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, Prentice Hall, Englewood Cliffs, 1989
[3] P.L. Combettes, J. Eckstein, Asynchronous block-iterative primal-dual decomposition methods for monotone inclusions, Math. Program. 168 (2018), 645-672.
[4] L. Elsner, I. Koltracht, M. Neumann, Convergence of sequential and asynchronous nonlinear paracontractions, Numer. Math. 62 (1992), 305-319.
[5] T. Huang, et al., A CNN-based policy for optimizing continuous action control by learning state sequences, Neurocomputing, 468 (2022), 286-295.
[6] L. Liu, et al., An asynchronous parallel stochastic coordinate descent algorithm, In: Proceedings of the 31st International Conference on Machine Learning (ICML-14), pp. 469-477, 2014.
[7] A. Frommer, D.B. Szyld, On asynchronous iterations, J. Comput. Appl. Math. 123 (2000), 201-216.
[8] G.M. Baudet, Asynchronous iterative methods for multiprocessors, J. ACM 25 (1978), 226-244.
[9] Z. Peng, et al., ARock: an algorithmic framework for asynchronous parallel coordinate updates, SIAM J. Sci. Comput. 38 (2016), A2851-A2879.
[10] G. Stathopoulos, C.N. Jones, An inertial parallel and asynchronous forward-backward iteration for distributed convex optimization, J. Optim. Theory App. 182 (2019), 1088-1119.
[11] L.C. Ceng, et al. A modified inertial subgradient extragradient method for solving pseudomonotone variational inequalities and common fixed point problems, Fixed Point Theory 21 (2020), 93-108.
[12] P.L. Combettes, I. Yamada, Compositions and convex combinations of averaged nonexpansive operators, J. Math. Anal. Appl. 425 (2015), 55-70.
[13] X. Qin, S.Y. Cho, Convergence analysis of a monotone projection algorithm in reflexive Banach spaces, Acta Math. Sci. 37 (2017), 488-502.
[14] Y. Shehu, J.C. Yao, Weak convergence of two-step inertial iteration for countable family of quasi-nonexpansive mappings, Ann. Math. Sci. Appl. 7 (2022), 259-279.
[15] X. Qin, H. Zhou, Common fixed points of a pair of non-expansive mappings with applications to convex feasibility problems, Glasgow Math. J. 52 (2010), 241-252.
[16] H. Heaton, Y. Censor, Asynchronous sequential inertial iterations for common fixed points problems with an application to linear systems, J. Global Optim. 74 (2019), 95-119.
[17] Q.L. Dong, et al., MiKM: Multi–step inertial Krasnosel'skiĭ–Mann algorithm and its applications, J. Global Optim. 73 (2019), 801-824.
[18] Q.L. Dong, et al., Convergence of projection and contraction algorithms with outer perturbations and their applications to sparse signals recovery, J. Fixed Point Theory Appl. 20 (2020), 16.
[19] P.C. Hansen, J.S. Jørgensen, AIR Tools II: algebraic iterative reconstruction methods, improved implementation, Numer. Algor. 79 (2018), 107-137.