Asynchronous Sequential Inertial Iterations for Common Fixed Points Problems

Presented by Howard Heaton[†]

Joint work with Yair Censor[*]

[†] *University of California Los Angeles*

[*] *University of Haifa*

December 13, 2018

Outline

UCLA

We explore the use of asynchrony for speeding up convergence when solving feasibility problems. The core idea is to take an inherently sequential process and express it in a fashion that allows for parallel implementations to occur in practice.

Let $\{C_i\}_{i=1}^m$ be a finite collection of closed convex sets contained in a Hilbert space $\mathcal{H}$ with a common fixed point. The associated *convex feasibility problem* (CFP) is

$$\text{Find } x^\star \in C := \bigcap_{i=1}^m C_i,$$

where we assume $C \neq \emptyset$.

Note: For simplicity, we only consider the consistent case, although extensions can be made for inconsistent problems.

In this work, we consider a collection of operators $\{T_i\}_{i=1}^m$ in $\mathcal{H}$ for which we set

$$C_i := \text{fix}(T_i) \text{ for all } i \in \{1, 2, \ldots, m\}. \tag{1}$$

The associated *common fixed points problem* is

$$\text{Find } x^\star \in C = \bigcap_{i=1}^m \text{fix}(T_i).$$

UCLA

Asynchronous algorithms offer

- robustness to dropped network signals
- easier coordination of nodes
- better utilization of processing power
- potentially higher-level parallelization
- (possibly) faster and fewer iterations

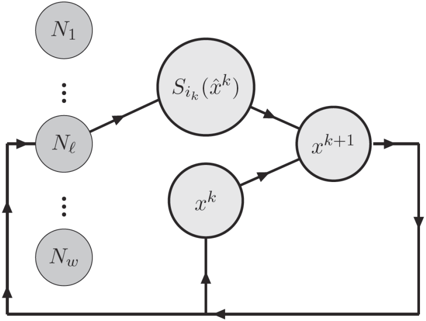

Figure 1: Schematic architecture model for ASI Algorithm. At the current iteration k, the latest output $N_\ell = S_{i_k}(\hat{x}^k)$ from the ℓ-th node is merged with x^k to form x^{k+1}, overwriting the global variable x^k. Here $w \le m$ is the number of nodes.

Async Notation

Suppose we have w processing nodes ($w \leq m$). For each time step k, let $d^k \in \mathbb{Z}^w_{\geq 0}$ be the delay vector. If the last output from the i-th node N_i was computing using x^{k-j}, then $(d^k)_i = j$. For a control sequence $\{i_k\} \subseteq \{1, 2, \ldots, m\}$ identifying the operator used to compute x^{k+1} we set $\hat{x}^k := x^{k-(d^k)_{i_k}}$.

Note: We have only proven convergence using *consistent reads*, and so we assume a queue is formed with locking when multiple updates arrive simultaneously.

Async Notation Example

If $k = 12$ is the current step, $i_{12} = 4$, and the last output was from the 4-th node and was generated using an iterate 3 steps out of date, then

$$(d^k)_{i_k} = (d^{12})_4 = 3, \tag{2}$$

and

$$\hat{x}^k = x^{k-(d^k)_{i_k}} = x^{12-3} = x^9. \tag{3}$$

Definition: Nonexpansive

An operator $T : \mathcal{H} \to \mathcal{H}$ is said to be nonexpansive provided

$$\|T(x) - T(y)\| \leq \|x - y\|, \quad \text{for all } x, y \in \mathcal{H}. \tag{4}$$

The operator S_i

Let $\{T_i\}_{i=1}^m$ be a collection of nonexpansive operators on $\mathcal{H}$ with a common fixed point. For each i set

$$S_i := \mathrm{Id} - T_i, \tag{5}$$

where Id is the identity operator.

Almost cyclic control

A sequence $\{i_k\}_{k\in\mathbb{N}}$ is called an *almost cyclic control* on $I := \{1, 2, \ldots, m\}$ if $\{i_k\} \subseteq I$ and there exists $K \geq m$ such that for each $k \in \mathbb{N}$ there is the containment $I \subseteq \{i_{k+1}, i_{k+2}, \ldots, i_{k+K}\}$.

Asynchronous Sequential Inertial (ASI) Algorithm

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

$$x^{k+1} := \begin{cases} x^k, & \text{if } k \leq \sup_{k\in\mathbb{N}} \|d^k\|_\infty, \\ x^k - \lambda_k S_{i_k}\left(\hat{x}^k\right), & \text{otherwise.} \end{cases} \tag{6}$$

Remark

The assignment of x^{k+1} to x^k for $k \leq \sup_{k\in\mathbb{N}} \|d^k\|_\infty$ is necessary to remove the possibility of having

$$\hat{x}^k = x^\ell, \tag{7}$$

with $\ell \leq 0$, which would be undefined since the iteration counter start at $k = 1$. In other words, the out-of-date iteration $\hat{x}^k$ cannot be more stale than the number of iterations k that have taken place.

The computation of x^{k+1} is expressible in two parts. Observe

$$\begin{aligned} x^{k+1} &= x^k - \lambda_{i_k} S_{i_k}\left(\hat{x}^k\right) \\ &= x^k - \lambda_{i_k}\left(\hat{x}^k - T_{i_k}\left(\hat{x}^k\right)\right) \\ &= \underbrace{(1-\lambda_{i_k})\, x^k + \lambda_{i_k} T_{i_k}\left(\hat{x}^k\right)}_{\text{convex combination}} + \underbrace{\lambda_{i_k}\left(x^k - \hat{x}^k\right)}_{\text{inertial term}}. \end{aligned} \tag{8}$$

Elsner et al. [2] in 1992 proved this iteration (without the inertial term) converges for any choice of λ_i's with $\lambda_i \in (0,1)$ when using paracontractions in finite dimensions. Referring to figure below, they set $x^{k+1} = y^k$.

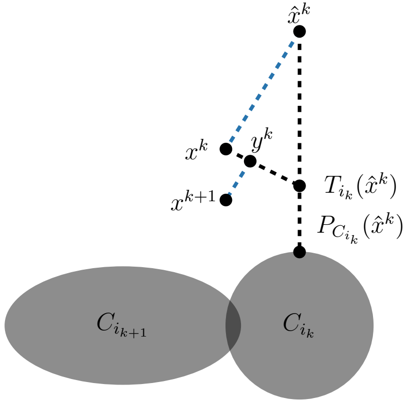

Figure 2: Illustration of a step of the ASI Algorithm with two convex sets and the T_i's as relaxed projections onto the sets. (Note the blue segments are parallel and the length of the lower is scaled by λ_k.)

In what follows, assume $\{T_i\}_{i=1}^m$ is a finite family of nonexpansive operators with a common fixed point and $C := \bigcap_{i=1}^m \mathrm{Fix}(T_i)$.

Lemma 1: Cluster Points are Fixed Points

Let $y \in \mathcal{H}$ 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 $y \in C$.

Proposition 1: Weak Convergence

Let $\{x^k\}_{k\in\mathbb{N}}$ be a sequence in $\mathcal{H}$. If for all $z \in C$ the sequence $\{\|x^k - z\|\}_{k\in\mathbb{N}}$ converges and if $\|T_i x^k - x^k\| \to 0$ for all $i \in [m]$, then the sequence $\{x^k\}_{k\in\mathbb{N}}$ converges weakly to a point $x^\star \in C$.

We henceforth assume $\{x^k\}_{k\in\mathbb{N}}$ is a sequence generated by the ASI algorithm and the delay vectors $\{d^k\}_{k\in\mathbb{N}}$ are uniformly bounded by some $\tau \geq 0$, i.e.,

$$\tau := \sup_{k\in\mathbb{N}} \|d^k\|_\infty < \infty. \tag{9}$$

Remark

The classical error $\|x^k - z\|$ is *not* necessarily nonincreasing, i.e., it is possible that there exists an index ℓ for which

$$\|x^{\ell+1} - z\| > \|x^\ell - z\|. \tag{10}$$

Lemma 2: A Fundamental Inequality

Let $z \in C$ and $\mu > 0$. Then

$$\begin{aligned} \|x^{k+1} - z\|^2 \leq \|x^k - z\|^2 + \mu \sum_{\ell=1}^{\tau} \|x^{k+1-\ell} - x^{k-\ell}\|^2 \\ - \lambda_k \left[1 - \lambda_k \left(1 + \tau/\mu\right)\right] \|S_{i_k}(\hat{x}^k)\|^2. \end{aligned} \tag{11}$$

Note: It is in the proof of this lemma where we utilize the fact T_{i_k} is nonexpansive, and so $\frac{1}{2}S_{i_k} = \frac{1}{2}(\mathrm{Id} - T_{i_k})$ is firmly nonexpansive.

Lemma 3: Lyapunov convergence

If $z \in C$, and $\mu > 0$, and $\varepsilon > 0$ is such that

$$0 < \varepsilon \leq \lambda_k \leq \frac{1}{1 + \tau(1/\mu + \mu) + \varepsilon}, \quad \text{for all } k \in \mathbb{N}, \tag{12}$$

then the sequence $\{\xi_k\}_{k \in \mathbb{N}}$ defined by

$$\xi_k := \|x^k - z\|^2 + \sum_{\ell=1}^{\tau} c_\ell \|x^{k+1-\ell} - x^{k-\ell}\|^2, \quad \text{for all } k \in \mathbb{N}, \tag{13}$$

converges, where

$$c_j := (\tau + 1 - j)\mu + \varepsilon, \quad \text{for all } j \in [\tau + 1]. \tag{14}$$

Lemma 4

If the assumptions of Lemma 3 hold, then $\|x^{k+1} - x^k\| \to 0$ and $\{\|x^k - z\|\}_{k\in\mathbb{N}}$ converges.

Lemma 5

If the assumptions of Lemma 3 hold, then $\|x^k - \hat{x}^k\| \to 0$.

Lemma 6

If the assumptions of Lemma 3 hold, then $\|T_i x^k - x^k\| \to 0$.

UCLA

Convergence Theorem

Let $\{x^k\}_{k\in\mathbb{N}}$ be generated by the ASI algorithm and the operators $\{T_i\}_{i=1}^m$ be nonexpansive. If the delay vectors are uniformly bounded in sup norm by some $\tau \geq 0$ and there is $\varepsilon > 0$ such that

$$0 < \varepsilon \leq \lambda_k \leq \frac{1}{2\tau + 1 + \varepsilon} \quad \text{for all } k \in \mathbb{N}, \tag{15}$$

then the sequence $\{x^k\}_{k\in\mathbb{N}}$ converges weakly to a solution $x^\star$, i.e.,

$$x^k \rightharpoonup x^\star \in C = \bigcap_{i=1}^m \text{fix}(T_i). \tag{16}$$

Outline

UCLA

Let $A \in \mathbb{R}^{M\times N}$ and $b \in \mathbb{R}^M$. Then for the i-th row of A, denoted a^i, define the hyperplane

$$H_i := \left\{ x \in \mathbb{R}^n : \langle a^i, x \rangle = b_i \right\}, \tag{17}$$

where $\langle \cdot, \cdot \rangle$ is the usual scalar product on $\mathbb{R}^n$. Finding $x^\star$ such that $Ax^\star = b$ is equivalent to having

$$x^\star \in \bigcap_{i=1}^{M} H_i. \tag{18}$$

Define operators T_i such that $\text{fix}(T_i) = H_i$ or some combination of the H_i's.

The Kaczmarz/ART method is defined via the iteration

$$\begin{aligned} x^{k+1} &= (1-\lambda_k)x^k + \lambda_k P_{i_k}\left(x^k\right) \\ &= x^k - \lambda_k \left(\frac{\langle a^{i_k}, x^k\rangle - b_{i_k}}{\|a^{i_k}\|^2} \right) a^{i_k}. \end{aligned} \tag{19}$$

Since the projection operators $\{P_i\}_{i=1}^M$ are nonexpansive, we may use the ASI algorithm framework to deduce the iteration

$$x^{k+1} = x^k - \lambda_k \left(\frac{\langle a^{i_k}, \hat{x}^k\rangle - b_{i_k}}{\|a^{i_k}\|^2} \right) a^{i_k} \tag{20}$$

also converges to a solution. We call this ASI-ART.

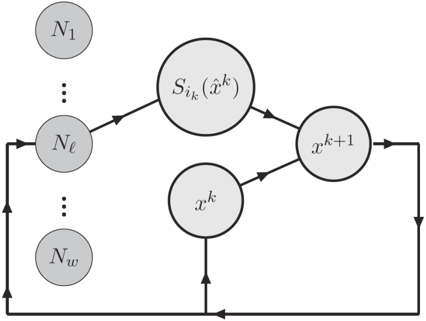

Figure 3: Schematic architecture for ASI Algorithm. At the current iteration k, the latest output $N_\ell = S_{i_k}(\hat{x}^k)$ from the ℓ-th node is merged with x^k to form x^{k+1}, overwriting the global variable x^k. Here $w \leq m$ is the number of nodes.

The fully simultaneous version of ART is Cimmino's method, defined by the update

$$x^{k+1} = x^k - \frac{\lambda_k}{M} \sum_{i=1}^{M} \frac{\langle a^i, x^k \rangle - b_i}{\|a^i\|^2} a^i. \tag{21}$$

Although the average of all the projections may be desirable, the $1/M$ term severely limits the speed of convergence. However, when A is sparse, the number of nonzero entries s_j in the j-th column of A satisfy $0 < s_j \ll M$. Censor et al. [1] (2008) proved convergence using Diagonally Relaxed Orthogonal Projections (DROP) where

$$x_j^{k+1} := x_j^k - \frac{\lambda_k}{s_j} \sum_{i=1}^{M} \frac{\langle a^i, x^k \rangle - b_i}{\|a^i\|^2} a_j^i \quad \text{for } j = 1, 2, \ldots, N. \qquad (22)$$

Define the matrices

$$D := \mathrm{diag}(1/s_j) \in \mathbb{R}^{N\times N} \quad \text{and} \quad W := \mathrm{diag}(1/\|a^i\|^2) \in \mathbb{R}^{M\times M}. \tag{23}$$

Then DROP becomes

$$x^{k+1} = x^k - \lambda_k D A^T W \left(Ax^k - b\right) \tag{24}$$

and

$$x^k \to x^* = \underset{x\in\mathbb{R}^N}{\arg\min} \|Ax - b\|_W. \tag{25}$$

For a family of block indices $\{B_t\}_{t=1}^r$, indicating subsets of the rows of A, we can associate the submatrix A_t with the rows in $t \in B_t$.

From each A_t, we may construct corresponding D_t, W_t, and b_t.

ASI-DROP Algorithm

Let $A \in \mathbb{R}^{M\times N}$, and $b \in \mathbb{R}^M$ be given. Choose $x^1 \in \mathbb{R}^N$, a sequence $\{\lambda_k\}_{k\in\mathbb{N}}$ such that $\lambda_k \in (0,1)$ for all $k \in \mathbb{N}$, an almost cyclic control $\{t_k\}_{k\in\mathbb{N}}$ on $[r]$, and an appropriate family of blocks of indices $\{B_t\}_{t=1}^r$. Then set

$$x^{k+1} := \begin{cases} x^k, & \text{if } k \leq \sup_{k\in\mathbb{N}} \|d^k\|_\infty, \\ x^k - \lambda_k D_{t_k} A_{t_k}^T W_{t_k} \left(A_{t_k}\hat{x}^k - b_{t_k}\right), & \text{otherwise.} \end{cases}$$

A Pseudocode Implementation of the ASI Algorithm

Initialization:

Let $x \in \mathcal{H}$, $\lambda \in (0,1)$, and $\{i_k\}_{k\in\mathbb{N}}$ an almost cyclic control on $[m]$. Set $k \leftarrow w+1$ and $\theta \leftarrow 1$.

for $\ell \in [w]$

Send x and i_ℓ to the ℓ-th node to compute $N_\ell = S_{i_\ell}(x)$

endfor

Master Node Iteration:

while stopping criteria not met

Fetch set of node indices F_θ for outputs received at time θ

for $\ell \in F_\theta$

$x \leftarrow x - \lambda N_\ell$,

$k \leftarrow k+1$

Send x and i_k to ℓ-th slave node to compute $N_\ell = S_{i_k}(x)$

endfor

$\theta \leftarrow \theta + 1$

end while

Slave Node ℓ Iteration:

Read x and i_k as input

Compute $N_\ell = S_{i_k}(x)$

Output $N_\ell = S_{i_k}(x)$ to master node

We illustrate the ASI algorithm on a feasibility problem (n.b. also incorporating TV reduction drastically reduces the number of rows needed). A linear system is generated modeling fan beam 2D CT data with a 176,672 $\times$ 16,384 matrix.

Figure 4: 128x128 Shepp Logan Phantom

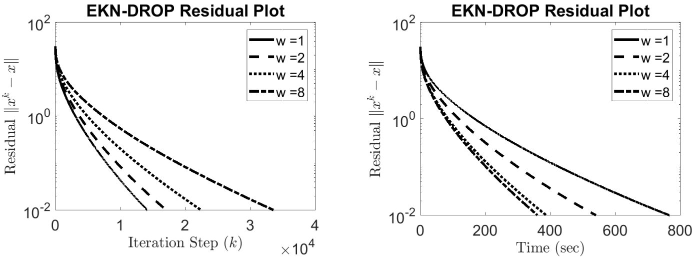

Figure 5: Averages of residual plots over 30 trials for EKN–DROP algorithm (without inertial terms).

Remark

Observe more iterations are needed as the number of nodes increases, yielding reduced efficiency and limited speedup.

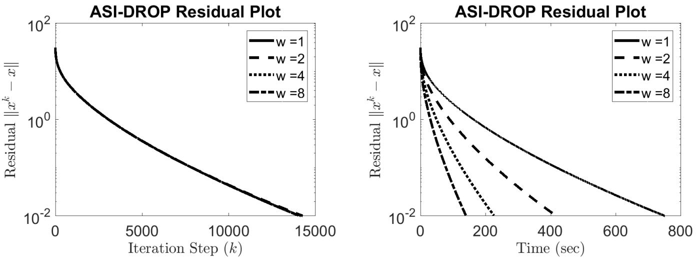

Figure 6: Averages of residual plots over 30 trials for ASI–DROP algorithm (with inertial terms).

Remark

Roughly the same number of iterations are needed as the number of nodes increases (when convergence is obtained). But, in our code, a queue forms with nodes waiting for the global variable x^k to update.

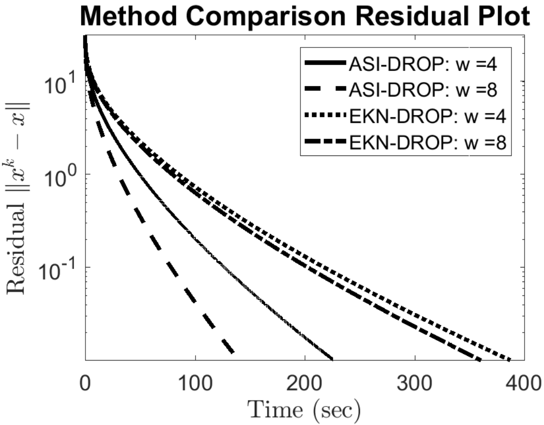

Figure 7: Comparison of methods plots (with/without inertial terms).

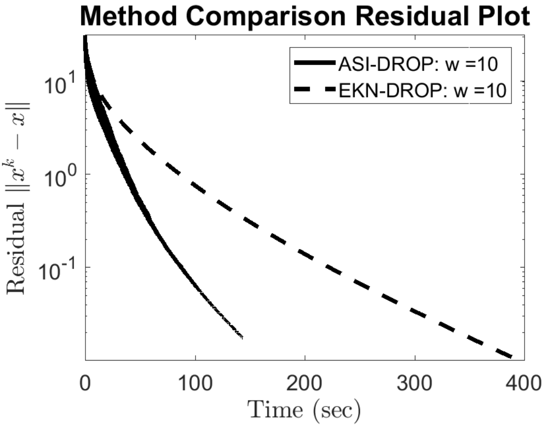

Figure 8: Comparison of methods plots (with/without inertial terms).

Small Model CT Problem Results

Here $m = 40$ operators T_i were used.

Method	Measurement	number of slave nodes (w)				
		$w = 1$	$w = 2$	$w = 4$	$w = 8$	$w = 10$
ASI–DROP	time (sec)	751.5	418.0	226.9	140.3	163.8
	# epochs	353.9	357.4	352.5	352.4	445.0
	speedup	NA	1.80	3.31	5.35	4.59
EKN–DROP	time (sec)	767.1	540.8	387.9	361.1	395.6
	# epochs	353.9	427.5	561.4	840.7	989.0
	speedup	NA	1.42	1.98	2.12	1.94

Table 1: Reconstruction results with iterations stopped when $\|x^k - x\| < \varepsilon = 10^{-2}$. Reported values are averaged from 30 trials repeated on the same data set.

Conclusions

- The ASI framework can be incorporated for robustness and speedup.
- Several algorithms occur as special cases of the ASI method (e.g., Kaczmarz's method, Cimmino's method, DROP, etc.).

[1] Yair Censor, Tommy Elfving, Gabor T Herman, and Touraj Nikazad.

On diagonally relaxed orthogonal projection methods.

SIAM Journal on Scientific Computing, 30(1):473–504, 2008.

[2] L. Elsner, I. Koltracht, and M. Neumann.

Convergence of sequential and asynchronous nonlinear paracontractions.

Numerische Mathematik, 62(1):305–319, Dec 1992.

[3] Robert Hannah and Wotao Yin.

On unbounded delays in asynchronous parallel fixed-point algorithms.

Journal of Scientific Computing, pages 1–28, 2016.

UCLA

[4] Zhimin Peng, Yangyang Xu, Ming Yan, and Wotao Yin.

ARock: An algorithmic framework for asynchronous parallel coordinate updates.

SIAM Journal on Scientific Computing, 38(5):A2851–A2879, 2016.