On the convergence rates of asynchronous iterations

Hamid Reza Feyzmahdavian and Mikael Johansson

***Abstract*— This paper presents a unifying convergence result for asynchronous iterations involving pseudo-contractions in the block-maximum norm. Contrary to previous results which only established asymptotic convergence or studied simplified models of asynchronism, our result allows to bound the convergence rates for both partially and totally asynchronous implementations. Several examples are worked out to demonstrate that our theorem recovers and improves on existing results, and that it allows to characterize the solution times for several classes of asynchronous iterations that have not been addressed before.**

I. INTRODUCTION

Asynchronous algorithms appear naturally in parallel and distributed systems and are heavily exploited applications ranging from large-scale linear algebra and optimization to distributed coordination of small embedded devices. Allowing nodes to operate in an asynchronous manner simplifies the implementation of distributed algorithms and eliminates the overhead associated with synchronization. However, care has to be taken, since asynchrony runs the risk of rendering an otherwise stable iteration unstable.

The dynamics of asynchronous iterations are much richer than their synchronous counterparts, and quantifying the impact of asynchrony on the convergence properties of iterative algorithms remains challenging. Some of the first results on the convergence of asynchronous iterations were derived by Chazan and Miranker [1], who studied chaotic relaxations for solving linear systems of equations. Several authors have proposed extensions of this pioneering work to nonlinear iterations involving maximum norm contractions (*e.g.*, [2], [3]) and for monotone iterations (*e.g.*, [4], [5]). Powerful convergence results for broad classes of asynchronous algorithms, including maximum norm contractions and monotone mappings, under different assumptions on communication delays and update rates were presented by Bertsekas [6] and Bertsekas and Tsitsiklis [7]. Most of the results in the literature only guarantee asymptotic convergence. This paper complements the existing work by developing convergence theorems that characterize the *rate* of convergence of asynchronous iterations and *quantify* how these rates depend on the update intervals and information delays in the system.

We focus on iterations involving block-maximum norm pseudo-contractions under the general asynchronous model introduced in [6], [7], which allows for heterogeneous and time-varying update rates and communication delays. Such iterations arise in a variety of algorithms, such as certain classes of linear fixed-point iterations and gradient descent methods [7], [8], optimum multiuser detection algorithms [9], distributed algorithms for averaging [10], and power control algorithms in wireless networks [11]–[13]. Our main theorem provides a powerful approach for characterizing the rate of convergence of totally asynchronous implementations, where both the update intervals and communication delays may grow unbounded. When specialized to partially asynchronous algorithms (where the update intervals and communication delays have a fixed upper bound), or to particular classes of unbounded delays and update intervals, our approach allows to explicitly quantify how the degree of asynchronism affects the convergence rates.

The paper is organized as follows. Section II reviews the partially and totally asynchronous models of computation and recalls some basic results about fixed-point iterations involving pseudo-contractions in the block-maximum norm. Section III presents our main results on the convergence rates of asynchronous iterations, and Section IV demonstrates how the results can be used to analyze the impact of asynchronism on the convergence rate of power control algorithms in wireless networks. Finally, Section V concludes the paper.

A. Notation and Preliminaries

Here, we introduce the notation and review the key definitions that will be used throughout the paper. We let $\mathbb{R}$, $\mathbb{N}$, and $\mathbb{N}_0$ denote the set of real numbers, natural numbers, and the set of natural numbers including zero, respectively. The largest integer less than or equal to real number x is indicated by $\lfloor x \rfloor$. The non-negative orthant of the n-dimensional real space $\mathbb{R}^n$ is represented by $\mathbb{R}^n_+$. For each vector $x = (x_1, \dots, x_m) \in \mathbb{R}^n$ with $x_i \in \mathbb{R}^{n_i}$, the *block-maximum norm* is defined by

$$\|x\|_b^w = \max_{1 \le i \le m} \frac{\|x_i\|_i}{w_i},$$

where w_i is a positive scalar, and $\|\cdot\|_i$ is a norm on $\mathbb{R}^{n_i}$. When $n_i = 1$ for all $i = 1, \dots, m$, the block-maximum norm reduced to the maximum norm defined by

$$\|x\|_\infty^w = \max_{1 \le i \le m} \frac{|x_i|}{w_i}.$$

A sequence $\{x(t)\} \in \mathbb{R}^n$ is said to converge *geometrically* (at a linear rate) to $x^\star$ if there exists a $\rho \in (0,1)$ such that

$$\lim_{t \to \infty} \frac{\|x(t+1) - x^\star\|}{\|x(t) - x^\star\|} = \rho,$$

where $\|\cdot\|$ is some norm on $\mathbb{R}^n$. For a matrix $A \in \mathbb{R}^{n \times n}$, a_{ij} denotes the entry in row i and column j. The spectral radius of A is the largest magnitude of its eigenvalues.

H. R. Feyzmahdavian and M. Johansson are with the Department of Automatic Control, School of Electrical Engineering and ACCESS Linnaeus Center, Royal Institute of Technology (KTH), SE-100 44 Stockholm, Sweden. Emails: {hamidrez, mikaelj}@kth.se.

II. Totally Asynchronous Algorithms involving Block-Maximum Norm Pseudo-Contractions

Consider an iterative algorithm on the form

$$x_i\big(t+1\big) = f_i\big(x_1(t),\ldots,x_m(t)\big), \quad t \in \mathbb{N}_0, \qquad (1)$$

where $i = 1,\ldots,m$, $x_i \in \mathbb{R}^{n_i}$, and $f_i : \mathbb{R}^n \to \mathbb{R}^{n_i}$ are functions of n variables with $n = n_1 + \ldots + n_m$. A vector $x^\star = (x_1^\star,\ldots,x_m^\star) \in \mathbb{R}^n$ is called a *fixed point* of the function $f(x) = (f_1(x),\ldots,f_m(x))$ if

$$x_i^\star = f_i(x_1^\star,\ldots,x_m^\star), \quad \forall i = 1,\ldots,m.$$

If f_i is continuous at $x^\star$ and the sequence $\{x_i(t)\}$ generated by (1) converges to $x_i^\star$ for every i, then $x^\star$ is a fixed point of f [7]. Therefore, the iteration (1) can be viewed as a network of m nodes, each responsible for updating one of the m subvectors of x so as to find a global fixed point. The function f is called a *pseudo-contraction* with respect to the block-maximum norm if there exists $c \in [0,1)$ such that

$$\|f(x) - x^\star\|_b^w \leq c\,\|x - x^\star\|_b^w, \quad \forall x \in \mathbb{R}^n,$$

where $x^\star$ is a fixed point of f. The scalar c is called the *contraction modulus* of f. Pseudo-contractions have at most one fixed point, to which the iterates produced by (1) converge geometrically [7].

The algorithm described by (1) is *synchronous* in the sense that all nodes update their states at the same time and have access to the states of all other nodes. Synchronous execution is possible if there are no communication faults or delays in the network and all nodes operate in synch with a global clock. In practice, these requirements are hard to satisfy: local clocks in different nodes tend to drift and communication latency between nodes can be significant and unpredictable. Synchronization can also be accomplished through communication primitives such as MPIs barrier, which enforces nodes to wait until all other nodes are ready to carry out the next iteration. The drawback with insisting on synchronous operation in an inherently asynchronous environment is that nodes will spend a significant time idle, especially if some nodes compute faster because of, *e.g.*, higher processor power or smaller workload per iteration.

In an *asynchronous* implementation of the iteration (1), each node updates its state at its own pace, using possibly outdated information from the other nodes. Following the notation in [7], we write such iterations as

$$x_i(t+1) = \begin{cases} f_i\big(x_1(\tau_1^i(t)),\cdots,x_m(\tau_m^i(t))\big), & t \in T^i, \\ x_i(t), & t \notin T^i, \end{cases} \qquad (2)$$

where T^i is the set of times when node i executes an update, and $\tau_j^i(t)$ is the time at which the most recent version of x_j available to node i at time t was computed. We can view $t - \tau_j^i(t)$ as the communication delay from node j to node i at time t. Note that $0 \leq \tau_j^i(t) \leq t$ for all $t \in \mathbb{N}_0$. The synchronous algorithm (1) is a special case of (2) where $\tau_j^i(t) = t$, and $T^i = \mathbb{N}_0$ for all i and j, and all $t \in \mathbb{N}_0$.

Based on the assumptions on the communication delays and update rates, asynchronous algorithms are classified into *totally asynchronous* and *partially asynchronous*:

Assumption 1 (Total Asynchronism [7]) *For the asynchronous algorithm* (2)*, there holds:*

a) *the sets T^i are infinite subsets of $\mathbb{N}_0$ for every i;*
b) $\lim_{k\to\infty} \tau_j^i(t_k) = \infty$ *for all i and j, where $\{t_k\}$ is a sequence of elements of T^i that tends to infinity.*

Loosely speaking, Assumption 1a) guarantees that no node ceases to execute its update while Assumption 1b) guarantees that old information is eventually purged out of the network. Under total asynchronism, the delay $t - \tau_j^i(t)$ can become unbounded as t increases. This is the main difference with partially asynchronous algorithms, where delays are assumed bounded; in particular, the following assumption holds.

Assumption 2 (Partial Asynchronism [7]) *For the asynchronous algorithm* (2)*, there exists a positive integer B such that:*

a) *For every i and for every $t \in \mathbb{N}_0$, at least one of the elements of the set $\{t, t+1, \ldots, t+B-1\}$ belongs to T^i.*
b) *There holds $0 \leq t - \tau_j^i(t) \leq B - 1$, for all i and j, and all $t \in \mathbb{N}_0$ belonging to T^i.*
c) *There holds $\tau_i^i(t) = t$ for all i and $t \in T^i$.*

Assumptions 2a) and 2b) ensure that both the time interval between updates executed by each node and the communication delays are bounded. When $B = 1$, this model reduces to the synchronous algorithm (1). Assumption 2c) states that nodes always use the latest version of their own state.

While convergent synchronous algorithms may diverge in the face of asynchronism, it has been shown in [7] that the asynchronous iteration (2) involving pseudo-contractions in the block-maximum norm also converges to the fixed point under total asynchronism, *i.e*, it can tolerate arbitrary large communication and computation delays. However, [7] did not quantify how bounds on the time delays and update rates of nodes affect the convergence rate of (2). One could expect that the convergence rates would become slower with increasing communication delays or with more infrequent update rates. Our main objective in this paper is therefore to give explicit estimates of the convergence rate of asynchronous algorithms involving block-maximum norm pseudo-contractions under different assumptions on communication delays and update rates.

III. Main Results

We will now develop a theorem that provides guaranteed convergence rates of the asynchronous algorithm (2) under various classes of total asynchronism. Our proof uses a continuous decreasing function $\lambda : \mathbb{R}_+ \to \mathbb{R}_+$ satisfying

$$\lim_{t\to\infty} \lambda(t) = 0,$$

and shows that for all $i = 1, \dots, m$, and for all $t \in \mathbb{N}_0$,

$$\frac{1}{w_i}\|x_i(t) - x_i^\star\|_i \leq M\lambda(t_k^i), \quad t \in (t_k^i, t_{k+1}^i],$$

where M is a positive constant, and t_k^i and t_{k+1}^i are two consecutive elements of T^i. The function $\lambda(t)$ quantifies how fast the sequence of vectors generated by (2) converges to the fixed point $x^\star$. For example, if $\lambda(t) = \rho^t$ with $\rho \in (0,1)$, $\{x_i(t_k^i)\}$ converges geometrically to $x_i^\star$; and if $\lambda(t) = t^{-\xi}$ with $\xi > 0$, then $\|x_i(t_k^i) - x_i^\star\|_i$ is upper bounded by a polynomial function of time. Similar to the asynchronous iterates themselves, the upper bound on the convergence rate is left unchanged when $t \notin T^i$ and decreases after update times; see Figure (1).

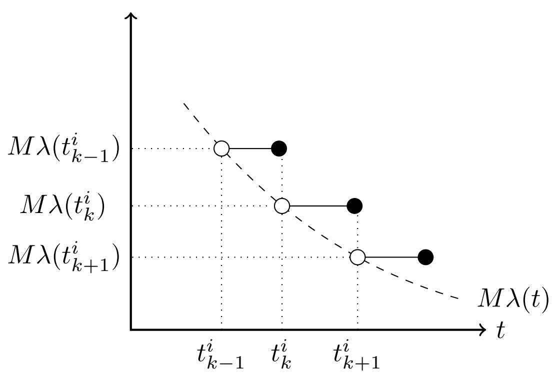

Fig. 1. Illustration of the upper bound on the convergence rate of the asynchronous algorithm (2) for every node i.

Theorem 1 *For the asynchronous algorithm* (2)*, suppose that the following conditions hold:*

i) f is a pseudo-contraction with contraction modulus c with respect to the block-maximum norm.

ii) There exist functions $\beta^i : \mathbb{R}_+ \to \mathbb{R}_+$ and $\Delta \in \mathbb{N}_0$ such that for all $t \geq \Delta$,

$$t - t_k^i \leq \beta^i(t) \leq t, \quad t \in (t_k^i, t_{k+1}^i], \tag{3}$$

where t_k^i and t_{k+1}^i are two consecutive elements of T^i.

iii) There is a decreasing function $\lambda : \mathbb{R}_+ \to \mathbb{R}_+$ such that

$$\lim_{t\to\infty} \lambda(t) = 0,$$

and that for all i and j,

$$c \lim_{t\to\infty} \frac{\lambda\big(\tau_j^i(t) - \beta^j(\tau_j^i(t))\big)}{\lambda(t)} < 1. \tag{4}$$

Then, the sequence of vectors generated by (2) *under total asynchronism satisfies*

$$\frac{1}{w_i}\|x_i(t) - x_i^\star\|_i \leq M\lambda(t_k^i), \quad t \in (t_k^i, t_{k+1}^i],$$

for all i and all $t \in \mathbb{N}_0$, where M is a positive constant.

Note that $\beta^i(t_{k+1})$ is an upper bound on the time interval between node i's kth and $k+1$st updates. Letting $\beta^i(t) = \beta$, $\beta \in \mathbb{N}$, means that node i performs at least one update during any time interval of length β. In general, $\beta^i(t)$ may be unbounded (we will will consider such a case in Example 1).

Proof: (*of Theorem 1*)

For each $i = 1, \dots, m$, let t_0^i be the first element of T^i. From Assumption 1b), there exists a time $\widehat{t} \in \mathbb{N}_0$ large enough such that for all i and j,

$$\tau_j^i(t) \geq \max\big\{\Delta, \max_{1\leq i\leq m}\{t_0^i\} + 1\big\}, \quad \forall t \geq \widehat{t}. \tag{5}$$

By (4), we can find a sufficiently large time $\widetilde{t} \in \mathbb{N}_0$ so that

$$c\lambda\big(\tau_j^i(t) - \beta^j(\tau_j^i(t))\big) \leq \lambda\big(t\big), \quad \forall t \geq \widetilde{t}. \tag{6}$$

Let $\bar{t} = \max\{\,\widehat{t}, \widetilde{t}\,\}$, and define

$$M = \frac{\|x(0) - x^\star\|_b^w}{\lambda(\bar{t})}.$$

According to Proposition 2.1 of Section 6.2 in [7], the sequence $\{x(t)\}$ generated by (2) satisfies

$$\frac{1}{w_i}\|x_i(t) - x_i^\star\|_i \leq \|x(0) - x^\star\|_b^w, \quad \forall t \in \mathbb{N}_0,$$

for all i. Thus,

$$\begin{aligned}\max_{0\leq t\leq \bar{t}}\left\{\frac{1}{w_i}\frac{\|x_i(t) - x_i^\star\|_i}{\lambda(t)}\right\} &\leq \max_{0\leq t\leq \bar{t}}\left\{\frac{\|x(0) - x^\star\|_b^w}{\lambda(t)}\right\}\\ &\leq \frac{\|x(0) - x^\star\|_b^w}{\lambda(\bar{t})}\\ &= M,\end{aligned}$$

where for the second inequality, we use the fact that $\lambda(t)$ is decreasing on $\mathbb{R}_+$. It follows that

$$\frac{1}{w_i}\|x_i(t) - x_i^\star\|_i \leq M\lambda(t), \quad \forall t \in \{0, \dots, \bar{t}\}.$$

For each $t_k^i \in T^i$, we have $\lambda(t) \leq \lambda(t_k^i)$ when $t \geq t_k^i$. Thus,

$$\frac{1}{w_i}\|x_i(t) - x_i^\star\|_i \leq M\lambda(t_k^i), \quad t \in (t_k^i, t_{k+1}^i], \tag{7}$$

for all $t \in \{0, \dots, \bar{t}\}$. We will show by induction that (7) also holds for all $t \geq \bar{t}$.

Assume for induction that (7) holds for all t up to some t', where $t' \geq \bar{t}$. Let $t_{k'}^i$ and $t_{k'+1}^i$ be two consecutive elements of T^i such that $t' \in (t_{k'}^i, t_{k'+1}^i]$. Using the induction hypothesis, we have

$$\frac{1}{w_i}\|x_i(t') - x_i^\star\|_i \leq M\lambda(t_{k'}^i). \tag{8}$$

We now prove that $x_i(t'+1)$ satisfies (7).

Case 1) If $t' \notin T^i$, then $t'+1 \in (t_{k'}^i, t_{k'+1}^i]$. Moreover, from (2), $x_i(t'+1) = x_i(t')$. It follows from (8) that

$$\frac{1}{w_i}\|x_i(t'+1) - x_i^\star\|_i = \frac{1}{w_i}\|x_i(t') - x_i^\star\|_i \leq M\lambda(t_{k'}^i).$$

Therefore, in this case, (7) is true for $t'+1$.

Case 2) If $t' \in T^i$, or, equivalently, $t' = t^i_{k'+1}$, then

$$\begin{aligned}\frac{1}{w_i}&\|x_i(t'+1) - x^\star_i\|_i \\ &= \frac{1}{w_i}\|f_i\big(x_1(\tau^i_1(t')),\cdots,x_m(\tau^i_m(t'))\big) - x^\star_i\|_i \\ &\leq c\big\|\big(x_1(\tau^i_1(t')),\cdots,x_m(\tau^i_m(t'))\big) - x^\star\big\|^w_b \\ &= c\max_{1\leq j\leq m}\left\{\frac{1}{w_j}\|x_j(\tau^i_j(t')) - x^\star_j\|_j\right\}, \qquad (9)\end{aligned}$$

where the inequality holds, since f is a pseudo-contraction with respect to the block-maximum norm. As $t' \geq \overline{t} \geq \widehat{t}$, (5) implies that $\tau^i_j(t') > t^j_0$ for each j. Let $t^j_{k_\tau}$ and $t^j_{k_\tau+1}$ be two consecutive elements of T^j such that

$$\tau^i_j(t') \in (t^j_{k_\tau}, t^j_{k_\tau+1}].$$

Since $\tau^i_j(t') \leq t'$, the induction hypothesis yields

$$\frac{1}{w_j}\|x_j(\tau^i_j(t')) - x^\star_j\|_i \leq M\lambda(t^j_{k_\tau}), \qquad (10)$$

for all j. Moreover, (5) also implies that $\tau^i_j(t') \geq \Delta$. It follows from (3) that

$$t^j_{k_\tau} \geq \tau^i_j\big(t'\big) - \beta^j\big(\tau^i_j(t')\big) \geq 0.$$

As $\lambda(t)$ is decreasing on R_+, this in turn implies

$$\lambda\big(t^j_{k_\tau}\big) \leq \lambda\big(\tau^i_j(t') - \beta^j(\tau^i_j(t'))\big). \qquad (11)$$

Substituting (10) into (9), then using (11), we obtain

$$\begin{aligned}\frac{1}{w_i}\|x_i(t'+1) - x^\star_i\|_i &\leq cM\max_{1\leq j\leq m}\lambda(t^j_{k_\tau}) \\ &\leq cM\max_{1\leq j\leq m}\lambda\big(\tau^i_j(t') - \beta^j(\tau^i_j(t'))\big) \\ &\leq M\lambda(t') \\ &= M\lambda(t^i_{k'+1}), \qquad (12)\end{aligned}$$

where the last inequality follows from (6). Note that

$$t'+1 = t^i_{k'+1} + 1 > t^i_{k'+1},$$

implying that $t'+1 \in (t^i_{k'+1}, t^i_{k'+2}]$. It follows from (12) that (7) holds for $t'+1$. The induction proof is complete. ∎

According to Theorem 1, any function $\lambda(t)$ satisfying condition (iii) can be used to estimate the convergence rate of the totally asynchronous algorithm (2). From (4), it is clear that the admissible choices for $\lambda(t)$ depend on the asymptotic behaviour of $\beta^i(t)$ and $\tau^i_j(t)$. This means that the rate at which the nodes execute their updates as well as the way communication delays tend large affects the convergence rate of (2). To clarify this statement, we will analyze a few special cases in detail. First, we consider the partially asynchronous model. The following result gives a bound on the convergence rate of asynchronous algorithms involving block-maximum norm pseudo-contractions under this model of asynchronicity.

Theorem 2 *Consider the iteration* (2) *under partial asynchronism. Assume that f is a block-maximum norm pseudo-contraction with contraction modulus c. Then, the sequence of vectors generated by* (2) *satisfies*

$$\frac{1}{w_i}\|x_i(t) - x^\star_i\|_i \leq M\rho^{t^i_k}, \quad t \in (t^i_k, t^i_{k+1}], \qquad (13)$$

for all i and all $t \in \mathbb{N}_0$, where M is a positive constant, t^i_k and t^i_{k+1} are two consecutive elements of T^i, and

$$\rho = c^{\frac{1}{2B-1}} \qquad (14)$$

Proof: According to Assumption 2a), we have

$$t - t^i_k \leq B \leq t, \quad t \in (t^i_k, t^i_{k+1}],$$

for all $t \geq B$. Thus, we can choose $\beta^i(t) = B$ for all i. Pick a constant $\hat{\rho}$ such that

$$\hat{\rho} \in (\rho, 1), \qquad (15)$$

where ρ is defined by (14). Let $\lambda(t) = \hat{\rho}^t$, $t \geq 0$. Clearly, $\lambda(t)$ is decreasing on $\mathbb{R}_+$. Moreover, for all i and j, we obtain

$$\begin{aligned}c\lim_{t\to\infty}\frac{\lambda\big(\tau^i_j(t) - \beta^j(\tau^i_j(t))\big)}{\lambda(t)} &= c\lim_{t\to\infty}\frac{\hat{\rho}^{\tau^i_j(t)-B}}{\hat{\rho}^t} \\ &\leq c\lim_{t\to\infty}\frac{\hat{\rho}^{t+1-2B}}{\hat{\rho}^t} \\ &< c\rho^{1-2B} = 1,\end{aligned}$$

where the first inequality uses the fact that under Assumption 2b), $t+1-B \leq \tau^i_j(t)$ for $t \in \mathbb{N}_0$. The last equality uses (14). It follows that condition (iii) of Theorem 1 holds for all $\hat{\rho}$ satisfying (15). Hence, the sequence $\{x(t)\}$ generated by (2) satisfies (13).

∎

Theorem 2 shows that block-maximum norm pseudo-contractions still converge geometrically under partial asynchronism assumption, and provides an explicit bound on the impact that an increasing delay has on the convergence rate. More precisely, $c^{1/(2B-1)}$ is monotonically increasing with B and approaches one as B tends to infinity. Hence, while the asynchronous algorithm (2) involving block-maximum norm pseudo-contractions remains geometrically stable for arbitrary bounded communication delays, the convergence rate deteriorates with increasing delays.

Contrary to the typical upper bounds on the convergence rate, the guaranteed bounds provided by Theorem 1 do not decrease at every time step, but only after update times t^i_k. Therefore, our estimation of convergence rate, in general, depends on how fast the sequence $\{t^i_k\}$ grows large. According to Theorem 2, the sequence $\{\|x_i(t^i_k) - x^\star_i\|_i\}$ generated by the partially asynchronous iteration (2) converges at a linear rate ρ. Under partial asynchronism, it holds that

$$0 \leq t - B \leq t^i_k, \quad t \in (t^i_k, t^i_{k+1}],$$

for all $t \geq B$, which implies that

$$M\rho^{t^i_k} \leq M\rho^{t-B} = M'\rho^t, \quad t \in (t^i_k, t^i_{k+1}],$$

where $M' = M\rho^{-B}$. It follows from (13) that

$$\frac{1}{w_i}\|x_i(t) - x_i^\star\|_i \leq M'\rho^t, \quad \forall t \geq B,$$

for all i. This shows that partially asynchronous iterations attains a rate of $O(\rho^t)$.

Under partial asynchronism, both update rates and communication delays are bounded. However, Theorem 1 can also be used to find guaranteed convergence rates of asynchronous iterations with unbounded communication delays and update intervals. To make our point, we establish convergence rates for a particular class of totally asynchronous algorithms described by the following assumption:

Assumption 3 *For the asynchronous algorithm* (2)*, there exist positive integer B, a scalar $\alpha \in [0, 1)$, and $t_\alpha \in \mathbb{N}_0$ such that, for each i and each $t \in T^i$, there holds:*

a) *There exists $t' \in T^i$ for which $1 \leq t' - t \leq B$.*
b) *$0 \leq t - \tau_j^i(t) \leq \alpha t$, for all $j \in \{1, \ldots, p\}$, and all $t \geq t_\alpha$.*

Note that delays satisfying Assumption 3b) may be unbounded (take, for example, $\tau_j^i(t) = \lfloor 0.2t \rfloor$, $t \in \mathbb{N}_0$). The associated convergence result now reads as follows.

Theorem 3 *Consider the iteration* (2) *under Assumption 3, and assume that f is a pseudo-contraction with contraction modulus c with respect to the block-maximum norm. Then, the sequence $\{x(t)\}$ generated by* (2) *satisfies*

$$\frac{1}{w_i}\|x_i(t) - x_i^\star\|_i \leq M\left(\frac{t_k^i}{B} + 1\right)^{-\xi}, \quad t \in (t_k^i, t_{k+1}^i], \quad (16)$$

for all i and all $t \in \mathbb{N}_0$, where M is a positive constant, t_k^i and t_{k+1}^i are two consecutive elements of T^i, and

$$\xi = \frac{\ln c}{\ln(1 - \alpha)}. \quad (17)$$

Proof: Similar to the proof of Theorem 2, we choose $\beta^i(t) = B$ for all $i = 1, \ldots, m$. Let

$$\lambda(t) = \left(\frac{t}{B} + 1\right)^{-\widehat{\xi}}, \quad t \geq 0,$$

where $\widehat{\xi}$ is a positive constant satisfying

$$\widehat{\xi} \in (0, \xi). \quad (18)$$

We then have

$$\begin{aligned} c \lim_{t\to\infty} \frac{\lambda\big(\tau_j^i(t) - \beta^j(\tau_j^i(t))\big)}{\lambda(t)} &= c \lim_{t\to\infty}\left(\frac{t/B + 1}{(\tau_j^i(t) - B)/B + 1}\right)^{\widehat{\xi}} \\ &\leq c \lim_{t\to\infty}\left(\frac{t + B}{(1-\alpha)t}\right)^{\widehat{\xi}} \\ &< \frac{c}{(1-\alpha)^{\xi}} = 1, \end{aligned}$$

where for the first inequality, we use the fact that

$$0 \leq (1 - \alpha)t \leq \tau_j^i(t), \quad t \geq t_\alpha.$$

The second inequality follows from (18). Therefore, according to Theorem 1, the sequence $\{x(t)\}$ generated by (2) satisfies (16). ■

According to Theorem 3, the convergence rate of the asynchronous algorithm (2) under unbounded delays satisfying Assumption 3 is upper bounded by a polynomial function of time. From (17), we can see that the magnitude of the upper delay bound, α, affects ξ. Specifically, ξ is monotonically decreasing with α and approaches zero as α tends to one. In addition, the upper bound on the convergence rate is inversely proportional to B. It follows that the convergence rates get increasingly slower as either delays are allowed to grow quicker when $t \to \infty$ or nodes execute less frequently.

The guaranteed bounds provided by Theorems 2 and 3 are derived under the assumption that the update intervals of all nodes are bounded by a constant B, *i.e.*,

$$t_{k+1}^i - t_k^i \leq B, \quad \forall k, i \quad (19)$$

However, Theorem 1 allows time-varying upper bounds on both update rates and communication delays. Rather than developing theorems for specific combinations of update rates and time delays, we illustrate the principle on a simple example.

Example 1 Consider the following asynchronous iteration

$$x(t+1) = \begin{cases} \frac{1}{2}x(t), & t \in T, \\ x(t), & t \notin T, \end{cases} \quad (20)$$

where $x(t) \in \mathbb{R}$, and $T = \{2^k \mid k \in \mathbb{N}_0\}$. In terms of (2), $f(x) = \frac{1}{2}x$. Note that f is a pseudo-contraction with $c = \frac{1}{2}$ and fixed point $x^\star = 0$. For any two consecutive elements of T, we have $t_{k+1} - t_k = 2^k$, $k \in \mathbb{N}_0$. Thus, there is no B satisfying (19). However, for all $t \in \mathbb{N}$,

$$t - t_k \leq \frac{1}{2}t \leq t, \quad t \in (t_k, t_{k+1}],$$

so (3) holds with $\beta(t) = t/2$. As $\lambda(t) = 1/t$ satisfies condition (iii) of Theorem 1, it follows that

$$|x(t)| \leq \frac{M}{t_k}, \quad t \in (t_k, t_{k+1}].$$

One can also verify that the sequence $\{x(t)\}$ generated by (20) is given by

$$x(t) = \frac{x(0)/2}{t_k}, \quad t \in (t_k, t_{k+1}],$$

for all $t \geq 2$. This shows that, in this example, both the iteration (20) and our guaranteed upper bound have the same convergence rate.

As also stressed in [14], very few results on convergence rates of asynchronous algorithms have appeared in the literature (see *e.g.*, [2], [7] for exceptions). In particular, [7, Section 6.3] showed that if delays are bounded and $T^i = \mathbb{N}_0$ for all i ($t_{k+1}^i - t_k^i = 1$, $\forall i, k$), then asynchronous iterations involving block-maximum norm pseudo-contractions converge geometrically to the fixed point. Theorems 2 and 3 as well as Example 1 demonstrate that not only can Theorem 1 recover the results in [7], but it also quantifies the convergence rates of asynchronous iterations with unbounded upper bounds on update intervals and communication delays.

IV. Asynchronous Optimization Algorithm for Power Control in Wireless

Next, we will use our main results to analyze the convergence of asynchronous power control algorithms in wireless networks. To this end, consider a wireless network where n mobile users communicate over the same frequency band. Since concurrent transmissions interfere with each other, users must transmit with sufficient power to overcome the interference caused by the others. However, increasing the transmit power of an individual user will not only increase its own power consumption (and hence drain the battery of the device quicker), but it will also generate more interference to the other users. Thus, a natural design goal is to minimize the total power consumption while guaranteeing that all users overcome the interference caused by the others. The optimal power allocation is then the one that solves the problem:

$$\begin{aligned} \min_{p} \quad & p \\ \text{subject to} \quad & p_i \geq I_i(p), \quad \text{for all } i = 1, \ldots, n. \end{aligned} \tag{21}$$

Here, $p = (p_1, \ldots, p_n) \in \mathbb{R}^n$, $p_i \in \mathbb{R}$ is the transmit power of user i, and $I_i(p)$ is the *interference function* modeling the effective interference of other users that user i must overcome. The definition of $I_i(p)$ depends on the communication technology, network configuration and user requirements; see *e.g.* [11], [15] for a wide range of examples. One of the simplest interference functions is the linear one, given by

$$I_i(p) = \gamma_i \frac{\sum_{j \neq i} g_{ij} p_j + h_i}{g_{ii}}, \tag{22}$$

where $g_{ij} \geq 0$ is the channel gain between user j and the receiver of user i, γ_i is the target Signal-to-Interference-and-Noise Ratio (SINR) of user i, and h_i is the background noise at the receiver of user i.

Linear and several important nonlinear interference functions share common properties that allow them to be analyzed in the framework of *contractive interference functions*.

Definition 1 ([11]) *A function $I : \mathbb{R}^n_+ \to \mathbb{R}^n_+$ is said to be a c-contractive interference function if for all $p \geq 0$ and for all $i = 1, \ldots, n$, it satisfies the following conditions:*

- *Positivity: $I_i(p) > 0$.*
- *Monotonicity: If $p \geq p'$, then $I_i(p) \geq I_i(p')$.*
- *Contractivity: There exists a constant $c \in [0, 1)$, and a vector $v > 0$ such that for all $\epsilon > 0$,*

$$I_i(p + \epsilon v) \leq I_i(p) + c\epsilon v_i.$$

Contractive interference functions are contractions (and hence pseudo-contractions) w.r.t. the maximum norm [11]. Moreover, when the interference function $I(p)$ is contractive, the optimization problem (21) is feasible, and its unique solution is given by the fixed point of the iteration

$$p_i(t+1) = I_i(p(t)), \quad t \in \mathbb{N}_0, \tag{23}$$

where $i = 1, \ldots, n$ [11]. The computation of the optimal transmit power by this iteration is simpler than using traditional Lagrangian methods, since no dual variables need to be stored and manipulated. Each user is only required to update its transmit power at every time step, using information of the transmit powers used by all users in the previous iteration.

In real-world networks, communication delays are inventible, and clock drift may cause some users to execute more iterations than others. When communication delays and asynchronous execution are accounted for, the power control algorithm (23) becomes

$$p_i(t+1) = \begin{cases} I_i\big(p_1(\tau_1^i(t)), \cdots, p_n(\tau_n^i(t))\big), & t \in T^i, \\ p_i(t), & t \notin T^i. \end{cases} \tag{24}$$

Since contractive interference functions are pseudo-contractions with respect to the maximum norm, Theorem 1 allows us to quantify the convergence rate of (24) under different classes of communication delays and update rates. Consider, for example, a situation where all mobiles update their powers at least once during any interval of length B, and there exists a positive integer $D_{\max}$ such that

$$t - D_{\max} \leq \tau_j^i(t) \leq t, \quad t \in \mathbb{N}_0, \tag{25}$$

holds for all i and j. The following result gives a bound on the convergence rate of (24) under assumptions above.

Corollary 1 *If $I(p)$ is c-contractive, then the asynchronous power control algorithm* (24) *satisfies*

$$\frac{1}{v_i}|p_i(t) - p_i^\star| \leq M\rho^{t_k^i}, \quad t \in (t_k^i, t_{k+1}^i], \tag{26}$$

for all $i = 1, \ldots, n$, and all $t \in \mathbb{N}_0$, where M is a positive constant, t_k^i and t_{k+1}^i are two consecutive elements of T^i, and $\rho = c^{\frac{1}{B+D_{\max}}}$.

In [15], it has been shown that for a class of interference functions, called *standard interference functions*, the asynchronous power control algorithm (24) converges asymptotically to the optimal power vector even when it is executed totally asynchronously. However, the impact of the communication delay and the update rate on the convergence rate of (24) has been missing in [15]. Several important standard interference functions proposed in the literature (for example, linear, macro diversity and minimum power assignment interference functions) are also contractive [11]. In [11], the convergence rate of asynchronous power control algorithms involving contractive interference functions was investigated under the assumption that all mobile users update their powers at each iteration ($T^i = \mathbb{N}$, for all i) and the communication delay is guaranteed to be bounded. In contrast, this paper develops tools that allow to quantify the convergence rate of (24) under various assumptions on communication delays and update rates. Specifically, Corollary 1 shows that (24) converges geometrically if the communication delays and update rates are bounded. An analogue corollary of Theorem 3 would demonstrate that the convergence rate of (24) is upper bounded by a polynomial function of time if Assumption 3 holds.

The following numerical example illustrates the accuracy of our guaranteed bounds on the convergence rate of asynchronous power control algorithms.

Example 2 We consider the asynchronous power control algorithm (24) with linear interference functions. Four mobile users share a channel with link gain matrix $G = [g_{ij}]$ where

$$G = \begin{bmatrix} 0.4000 & 0.0082 & 0.0419 & 0.0579 \\ 0.0160 & 0.8530 & 0.0424 & 0.0043 \\ 0.0200 & 0.0017 & 0.1405 & 0.0010 \\ 0.1030 & 0.0036 & 0.0104 & 0.4050 \end{bmatrix} \times 10^{-3}.$$

The SINR threshold and the background noise for each user is set to $\gamma_i = 3$ and $h_i = 0.04$ mWatts, respectively. Let $\overline{G} = [\overline{g}_{ij}]$ be an 4×4 matrix with $\overline{g}_{ii} = 0$ and $\overline{g}_{ij} = \gamma_i g_{ij}/g_{ii}$ for $j \neq i$. Since the spectral radius of $\overline{G}$ is $0.7146 < 1$, the linear interference function is 0.7146-contractive with respect to the maximum norm $\|\cdot\|_\infty^v$, where $v = (0.59, 0.14, 0.38, 0.67)^T$ is the right Perron-Frobenius eigenvector of $\overline{G}$ [11].

To demonstrate the flexibility of our framework, assume that each user i executes (24) under the assumptions that:

- $T^i = \{ik \mid k \in \mathbb{N}_0\}$;
- $\tau_i^i(t) = 0$, for all i and all $t \in \mathbb{N}_0$;
- For all i and j with $j \neq i$,

$$\tau_i^j(t) = \begin{cases} t, & 0 \leq t \leq 4, \\ t - 0.5j\big(1 + (-1)^t\big), & 5 \leq t. \end{cases}$$

It is easy to verify that the time interval between any two consecutive updates executed by all nodes is upper bounded by $B = 4$, and (25) holds with $D_{\max} = 4$. Therefore, according to Corollary 1, the asynchronous algorithm (24) converges geometrically to the unique fixed point. In particular, the transmit power of each user satisfies (26) with

$$\rho = (0.7146)^{\frac{1}{8}} = 0.9588.$$

Figure 2 gives the simulation results of the theoretical bound obtained from Corollary 1 and the actual convergence rate of (24) for users 3 and 4.

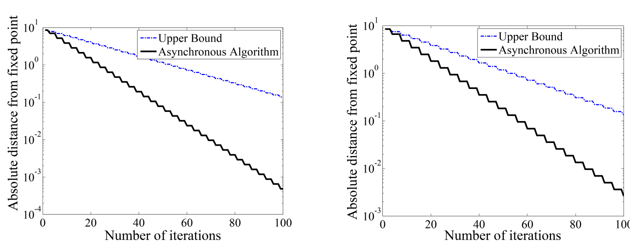

Fig. 2. Upper bound and actual convergence rate of (24) for user 3 (left) and user 4 (right) in the wireless network described in Example 2. The horizontal axis represents the number of iterations and the vertical axis shows $\frac{1}{v_i}|p_i(t) - p_i^\star|$, $i = 3, 4$ (in logarithmic scale).

V. Conclusions and Future Directions

This paper presented a convergence result for asynchronous iterations involving pseudo-contractions in the block-maximum norm. Contrary to most results in the literature, our theorems allow to characterize the *rates* of convergence of asynchronous iterations and *quantify* how these rates depend on the update intervals and information delays in the system. We demonstrated how our results can be used to analyze the impact of asynchrony on the convergence rate of power control algorithms in wireless networks.

There are several open issues for future work, such as attempting to derive convergence rates of asynchronous iterations involving monotone mappings [16], pseudo-contractions with respect to the Euclidean norm [17], and non-expansive mappings [18], much as was done in [19] for the case of non-expansive linear iterations with delays.

References

[1] D. Chazan and W. Miranker, "Chaotic relaxation," *Linear algebra and its applications*, vol. 2, no. 2, pp. 199–222, 1969.

[2] G. M. Baudet, "Asynchronous iterative methods for multiprocessors," *Journal of the ACM (JACM)*, vol. 25, no. 2, pp. 226–244, 1978.

[3] M. N. El Tarazi, "Some convergence results for asynchronous algorithms," *Numerische Mathematik*, vol. 39, no. 3, pp. 325–340, 1982.

[4] D. P. Bertsekas, "Distributed dynamic programming," *IEEE Transactions on Automatic Control*, vol. 27, no. 3, pp. 610–616, 1982.

[5] D. P. Bertsekas and D. El Baz, "Distributed asynchronous relaxation methods for convex network flow problems," *SIAM Journal on Control and Optimization*, vol. 25, no. 1, pp. 74–85, 1987.

[6] D. P. Bertsekas, "Distributed asynchronous computation of fixed points," *Mathematical Programming*, vol. 27, pp. 107–120, 1983.

[7] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation*. Prentice-Hall, 1989.

[8] C. C. Moallemi and B. V. Roy, "Convergence of min-sum message-passing for convex optimization," *IEEE Transactions on Information Theory*, vol. 56, pp. 2041–2050, 2010.

[9] A. Yener, R. D. Yates, and S. Ulukus, "CDMA multiuser detection: A nonlinear programming approach," *IEEE Transactions on Communications*, vol. 50, no. 6, pp. 1016–1024, 2002.

[10] M. Mehyar, D. Spanos, J. Pongsajapan, S. H. Low, and R. M. Murray, "Asynchronous distributed averaging on communication networks," *IEEE/ACM Transactions on Networking*, vol. 15, pp. 512–520, 2007.

[11] H. Feyzmahdavian, M. Johansson, and T. Charalambous, "Contractive interference functions and rates of convergence of distributed power control laws," *IEEE Transactions on Wireless Communications*, vol. 11, no. 12, pp. 4494–4502, Dec. 2012.

[12] H. R. Feyzmahdavian, T. Charalambous, and M. Johansson, "Asymptotic and exponential stability of general classes of continuous-time power control laws in wireless networks," *52nd IEEE Conference on Decision and Control (CDC)*, pp. 49–54, 2013.

[13] ——, "Stability and performance of continuous-time power control in wireless networks," *IEEE Transactions on Automatic Control*, vol. 59, no. 8, pp. 2012–2023, 2014.

[14] H. Avron, A. Druinsky, and A. Gupta, "Revisiting asynchronous linear solvers: Provable convergence rate through randomization," *IPDPS*, 2014, Available: *http://arxiv.org/abs/1304.6475*.

[15] R. Yates, "A framework for uplink power control in cellular radio systems," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 7, pp. 1341–1347, 1995.

[16] D. P. Bertsekas and H. Yu, "Distributed asynchronous policy iteration in dynamic programming," *48th Annual Allerton Conference on Communication, Control, and Computing*, pp. 1368–1375, 2010.

[17] H. H. Bauschke and P. L. Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, 2011.

[18] P. Tseng, D. P. Bertsekas, and J. N. Tsitsiklis, "Partially asynchronous, parallel algorithms for network flow and other problems," *SIAM Journal on Control and Optimization*, vol. 28, pp. 678–710, 1990.

[19] A. Nedić and A. Ozdaglar, "Convergence rate for consensus with delays," *Journal of Global Optimization*, vol. 47, pp. 437–456, 2010.