Asynchronous Distributed Optimization using a Randomized Alternating Direction Method of Multipliers

Franck Iutzeler, Pascal Bianchi, Philippe Ciblat and Walid Hachem

***Abstract*— Consider a set of networked agents endowed with private cost functions and seeking to find a consensus on the minimizer of the aggregate cost. A new class of random asynchronous distributed optimization methods is introduced. The methods generalize the standard Alternating Direction Method of Multipliers (ADMM) to an asynchronous setting where isolated components of the network are activated in an uncoordinated fashion. The algorithms rely on the introduction of *randomized* Gauss-Seidel iterations of a Douglas-Rachford operator for finding zeros of a sum of two monotone operators. Convergence to the sought minimizers is provided under mild connectivity conditions. Numerical results sustain our claims.**

I. Introduction

Consider a network represented by a set V of agents seeking to solve the following optimization problem on a Euclidean space X:

$$\inf_{x\in\mathsf{X}} \sum_{v\in V} f_v(x) \ , \tag{1}$$

where f_v is a convex real function known by agent v only. Function f_v can be interpreted as the price payed by an agent v when the global network state is equal to x.

This problem arises for instance in *cloud learning* applications where massive data sets are distributed in a network and processed by distinct virtual machines [1]. We investigate distributed optimization algorithms: agents iteratively update a local estimate using their private objective f_v and, simultaneously, exchange information with their neighbors in order to eventually reach a consensus on the global solution. Standard algorithms are generally *synchronous*: all agents are supposed to complete their local computations synchronously at each tick of an external clock, and then synchronously merge their local results. However, in many situations, one faces variable sizes of the local data sets along with heterogeneous computational abilities of the virtual machines. Synchronism then becomes a burden, as the global convergence rate is expected to depend on the local computation times of the slowest agents. It is crucial to introduce asynchronous methods which allow the estimates to be updated in a non-coordinated fashion, rather than all together or in some frozen order.

The literature contains at least three classes of distributed optimization methods for solving (1). The first one is based on the simultaneous use of a local *first-order* optimization algorithm (subgradient algorithm [2], [3], [4], Nesterov-like method [5], [6]) and a gossip process which drives the network to a consensus. A second class of methods is formed by distributed Newton-Raphson methods [7]. This paper focuses on a third class of methods derived from proximal splitting methods [8], [9], [10]. Perhaps the most emblematic proximal splitting method is the so-called *Alternating Direction Method of Multipliers* (ADMM) recently popularized to multiagent systems by the monograph [11]. Schizas *et al.* demonstrated the remarkable potential of ADMM to handle distributed optimization problems and introduce a useful framework to encompass graph-constrained communications [12]. We also refer to [13], [14] for recent contributions. However, all of these works share a common perspective: Algorithms are synchronous. They require a significant amount of coordination or scheduling between agents. In [12], [13], agents operate in parallel, whereas [14] proposes a sequential version of ADMM where agents operate one after the other in a predetermined order.

Contributions. This paper introduces a novel class of distributed algorithms to solve (1). The algorithms are asynchronous in the sense that some components of the network are allowed to wake up at random and perform local updates, while the rest of the network stands still. No coordinator or global clock is needed. The frequency of activation of the various network components is likely to vary. The algorithms rely on the introduction of *randomized* Gauss-Seidel iterations of a Douglas-Rachford monotone operator. We prove that the latter iterations provides a new powerful method for finding the zeros of a sum of two monotone operators. Application of our method to problem (1) yields a randomized ADMM-like algorithm, which is proved to converge to the sought minimizers.

The paper is organized as follows. The distributed optimization problem is rigorously stated in Section II. The synchronous ADMM algorithm that solves this problem is then described in Section III. Section IV forms the core of the paper. After quickly recalling the monotone operator formalism, the random Gauss-Seidel form of the proximal algorithm is described and its convergence is shown there. These results will eventually lead to an asynchronous version of the well-known Douglas-Rachford splitting algorithm. In Section V, the results of Section IV are applied towards developing an asynchronous version of the ADMM algorithm. An implementation example is finally provided in Section VI along with some simulations in Section VII.

Notations

Consider a non-directed graph $G = (V, E)$ where V is a set of vertices and E a set of edges. We sometimes note

This work was partially funded by the French Defense Agency (DGA).

The authors are with Telecom ParisTech, CNRS LTCI, 75013 Paris, France {lastname}@telecom-paristech.fr

$v \sim w$ for $\{v,w\} \in E$. For any $A \subset V$, we denote by $G(A)$ the subgraph of G induced by A (*i.e.*, $G(A)$ has vertices A and for any $(v,w) \in A^2$, $\{v,w\}$ is an edge of $G(A)$ if and only if it is an edge of G). Let X be a Euclidean space. We denote by X^A the set of functions on $A \to \mathsf{X}$. It is endowed with the inner product $\langle x,y\rangle_A = \sum_{v\in A}\langle x(v),y(v)\rangle_{\mathsf{X}}$ where $\langle\,.\,,\,.\,\rangle_{\mathsf{X}}$ is the inner product on X. We will omit subscripts $_{\mathsf{X}}$ and $_A$ when no confusion occurs. For any finite collection $A_1,\cdots,A_L \subset V$, we endow the space $\mathsf{X}^{A_1}\times\cdots\times\mathsf{X}^{A_L}$ with the scalar product $\langle x,y\rangle = \sum_{\ell=1}^L \langle x_\ell,y_\ell\rangle_{A_\ell}$ for any $x=(x_1,\cdots,x_L)$ and $y=(y_1,\cdots,y_L)$.

We denote by $\Pi_A x$ the restriction of x to A *i.e.*, $\Pi_A : \mathsf{X}^V \to \mathsf{X}^A$ is the linear operator defined for any $x \in \mathsf{X}^V$ as $\Pi_A x : (v \in A) \mapsto x(v)$. We denote by $1_A \in \mathsf{X}^A$ the constant function equal to one and by $\mathrm{sp}(1_A)$ the linear span of 1_A *i.e.*, the set of constant functions on A. Notation $|A|$ represents the cardinal of a set A.

For a closed proper convex function $h : \mathsf{X} \to (-\infty,+\infty]$ we define $\mathrm{prox}_{h,\rho}(x) = \arg\min_y h(y) + \frac{\rho}{2}\|y-x\|^2$.

II. Distributed Optimization on a Graph

Consider a network of agents represented by a non-oriented graph $G=(V,E)$ where V is a finite set of vertices (*i.e.*, the agents) and E is a set of edges. Each agent $v \in V$ has a private cost function $f_v : \mathsf{X} \to (-\infty,+\infty]$ where X is a Euclidean space. We make the following assumption on functions f_v.

Assumption 1:

i) For all $v \in V$, f_v is a proper closed convex function.

ii) The infimum in (1) is finite and is attained at some point $x^* \in \mathsf{X}$.

In order to solve the optimization problem (1) on the graph G, we first provide an equivalent formulation of (1) that will be revealed useful. For some integer $L \geq 1$, consider a finite collection $A_1, A_2, \cdots, A_L$ of subsets of V which we shall refer to as *components*. We assume the following condition.

Assumption 2: i) $\bigcup_{\ell=1}^L A_\ell = V$.

ii) $\bigcup_{\ell=1}^L G(A_\ell)$ is connected.

Assumption 2*i)* implies that any vertex appears in one of the components $A_1,\cdots,A_L$ at least. We stress the fact that two distinct components A_ℓ and $A_{\ell'}$ are not necessarily disjoint, though. Assumption 2*ii)* means that the union of all subgraphs is connected. As the latter union is also a subgraph of G, this implies that G is connected. As will be made clear below, our algorithms shall assume that all agents in the same component are able to perform simple operations in a coordinated fashion (*i.e.*, compute a local average over a component). Thus, in practice, it is reasonable to require that each subgraph $G(A_\ell)$ is itself connected.

We introduce some notations. We set for any $x \in \mathsf{X}^V$,

$$f(x) \triangleq \sum_{v\in V} f_v(x(v)) \ .$$

For any $z=(z_1,\cdots,z_L) \in \mathsf{Z} \triangleq \mathsf{X}^{A_1}\times\cdots\times\mathsf{X}^{A_L}$, we define the closed proper convex function

$$g(z) \triangleq \sum_{\ell=1}^L \iota_{\mathrm{sp}(1_{A_\ell})}(z_\ell)$$

where ι_H is the indicator function of a set H (equal to zero on H and to $+\infty$ outside). Here $g(z)$ is equal to zero if for any ℓ, z_ℓ is constant. Otherwise, $g(z)$ is infinite. For any $x \in \mathsf{X}^V$, we define $Mx \triangleq (\Pi_{A_1}x,\cdots,\Pi_{A_L}x)$. We consider the following optimization problem:

$$\inf_{x\in\mathsf{X}^V} \ f(x) + g(Mx) \tag{2}$$

Lemma 1: Under Assumption 2, x is a minimizer of (2) if and only if $x = \bar{x}1_V$ where $\bar{x} \in \mathsf{X}$ is a minimizer of (1).

Proof: Let $x \in \mathsf{X}^V$ such that $g(Mx)$ is finite. Then x is constant on each component. Let v,w be two arbitrary vertices in V. There exists a path in $\bigcup_{\ell=1}^L G(A_\ell)$ connecting v and w. Each edge of this path connects two vertices which belong to a common component. Thus, x is constant on two consecutive vertices of the path. This proves that $x(v) = x(w)$. Thus, x is constant and the result follows. ∎

As noted in [9], solving Problem (2) is equivalent to the search of the zeros of two monotone operators. One of possible approaches for that sake is to use ADMM. Although the choice of the sets $A_1,\cdots,A_L$ does not change the minimizers of the initial problem, it has an impact on the particular form of ADMM used to find these minimizers, as we shall see below.

In order to be more explicit, we provide in this section two important examples of possible choices for the components $A_1,\cdots,A_L$.

Example 1: Let $L=1$ and $A_1=V$. Problem (2) writes

$$\inf_{x\in\mathsf{X}^V} \ f(x) + \iota_{\mathrm{sp}(1_V)}(x) \ ,$$

In this case, the formulation is identical to [11, Chapter 7].

Example 2: Let $L=|E|$ and $\{A_1,\cdots,A_L\}=E$. That is, each set A_ℓ is a pair of vertices $\{v,w\}$ such that $\{v,w\}$ is an edge. Problem (2) writes

$$\inf_{x\in\mathsf{X}^V} \ f(x) + \sum_{v\sim w} \iota_{\mathrm{sp}(\mathbf{1}_2)}\begin{pmatrix}x(v)\\x(w)\end{pmatrix}$$

where $\mathbf{1}_2$ stands for the vector $(1,1)^T$.

III. Synchronous ADMM

A. General facts

We now apply the standard ADMM to Problem (2). Perhaps the most direct way to describe ADMM is to reformulate the unconstrained problem (2) into the following constrained problem: Minimize $f(x)+g(z)$ subject to $z=Mx$. For any $x \in \mathsf{X}^V$, $\lambda, z \in \mathsf{Z}$, the augmented Lagrangian is given by

$$\mathcal{L}_\rho(x,z;\lambda) \triangleq f(x)+g(z)+\langle\lambda, Mx-z\rangle+\frac{\rho}{2}\|Mx-z\|^2 \tag{3}$$

where $\rho>0$ is a constant. ADMM consists of the iterations

$$x^{k+1} = \operatorname*{argmin}_{x\in\mathsf{X}^V} \mathcal{L}_\rho(x,z^k;\lambda^k) \tag{4a}$$

$$z^{k+1} = \operatorname*{argmin}_{z\in\mathsf{Z}} \mathcal{L}_\rho(x^{k+1},z;\lambda^k) \tag{4b}$$

$$\lambda^{k+1} = \lambda^k + \rho\left(Mx^{k+1}-z^{k+1}\right). \tag{4c}$$

From [11, Chap. 3.2], the following result is immediate.

Theorem 1: Under Assumption 1, the sequence (x^k) defined in (4a) converges to a minimizer of (2).

B. Decentralized Implementation

One should now make (4) more explicit and convince the reader that the iterations are indeed amenable to distributed implementation. Due to the specific form of function g, it is clear from (4b) that all components $z_1^k, \cdots, z_L^k$ of z^k are constant. Otherwise stated, $z^k = (\bar{z}_1^k 1_{A_1}, \cdots, \bar{z}_L^k 1_{A_L})$ for some constants $\bar{z}_\ell^k \in \mathsf{X}$. For any $v \in V$, we set

$$\sigma(v) \triangleq \{\ell \,:\, v \in A_\ell\}\ .$$

Now consider the first update equation (4a). Getting rid of all quantities in $\mathcal{L}_\rho$ which do not depend on the vth component of x, we obtain for any $v \in V$

$$x^{k+1}(v) = \operatorname*{argmin}_{y \in \mathsf{X}} \ f_v(y) + \sum_{\ell \in \sigma(v)} \langle \lambda_\ell^k(v), y\rangle + \frac{\rho}{2}\|y - \bar{z}_\ell^k\|^2\,.$$

After some algebra, the above equation further simplifies to

$$x^{k+1}(v) = \mathrm{prox}_{f_v, \rho|\sigma(v)|}\left(Z^k(v) - B^k(v)\right) \tag{5}$$

where we introduced the following constants:

$$Z^k(v) = \frac{1}{|\sigma(v)|}\sum_{\ell \in \sigma(v)} \bar{z}_\ell^k, B^k(v) = \frac{1}{\rho|\sigma(v)|}\sum_{\ell \in \sigma(v)} \lambda_\ell^k(v)\,. \tag{6}$$

It is straightforward to show that the second update equation (4b) admits as well a simple decomposable form. After some algebra, we obtain that for any $\ell = 1, \cdots, L$,

$$\bar{z}_\ell^{k+1} = \frac{1}{|A_\ell|}\sum_{v \in A_\ell} x^{k+1}(v) + \frac{\lambda_\ell^k(v)}{\rho}\,. \tag{7}$$

Finally, for all $\ell = 1, \cdots, L$ and $v \in A_\ell$, equation (4c) reads

$$\lambda_\ell^{k+1}(v) = \lambda_\ell^k(v) + \rho(x^{k+1}(v) - \bar{z}_\ell^{k+1})\,. \tag{8}$$

Averaging (8) w.r.t. v and using (7) yields $\sum_{v \in A_\ell} \lambda_\ell^k(v) = 0$. Thus, the second term in the RHS of (7) can be deleted. Finally, averaging (8) w.r.t. ℓ leads to

$$B^{k+1}(v) = B^k(v) + x^{k+1}(v) - Z^{k+1}(v)\,. \tag{9}$$

Synchronous ADMM:
At each iteration k,
For each agent v, compute $x^{k+1}(v)$ using (5).
In each components $\ell = 1, \cdots, L$, compute

$$\bar{z}_\ell^{k+1} = \frac{1}{|A_\ell|}\sum_{w \in A_\ell} x^{k+1}(w).$$

For each agent v, compute $Z^{k+1}(v)$ and $B^{k+1}(v)$ using (6) and (9) respectively.

The above algorithm implicitly requires the existence of a routine for computing an average, in each component A_ℓ. This requirement is mild when the components coincide with edges of the graph as in Example 2. In this case, one only needs that the two vertices of an edge share their current estimate and find an agreement on the average. In the general case, the objective can be achieved by selecting a leader in each component whose role is to gather the estimates, compute the average and send the result to all agents in this component.

It is worth noting that in the case of Example 1, the synchronous ADMM described above coincides with the algorithm of [11].

IV. A Randomized Proximal Algorithm

A. Monotone operators

An operator T on a Euclidean space Y is a set valued mapping $\mathsf{T} : \mathsf{Y} \to 2^{\mathsf{Y}}$. An operator can be equivalently identified with a subset of $\mathsf{Y} \times \mathsf{Y}$, and we write $(x, y) \in \mathsf{T}$ when $y \in \mathsf{T}(x)$. Given two operators T_1 and T_2 on Y and two real numbers α_1 and α_2, the operator $\alpha_1\mathsf{T}_1 + \alpha_2\mathsf{T}_2$ is defined as $\alpha_1\mathsf{T}_1 + \alpha_2\mathsf{T}_2 = \{(x, \alpha_1 y_1 + \alpha_2 y_2) \,:\, (x, y_1) \in \mathsf{T}_1,\ (x, y_2) \in \mathsf{T}_2\}$. The identity operator is $\mathsf{I} = \{(x, x) : x \in \mathsf{Y}\}$ and the inverse of the operator T is $\mathsf{T}^{-1} = \{(x, y) \,:\, (y, x) \in \mathsf{T}\}$. The operator T is said *monotone* if

$$\forall\ (x, y), (x', y') \in \mathsf{T},\ \langle x - x', y - y'\rangle \geq 0.$$

A monotone operator is said *maximal* if it is not strictly contained in any monotone operator (as a subset of $\mathsf{Y} \times \mathsf{Y}$). Finally, T is said *firmly non-expansive* if

$$\forall\ (x, y), (x', y') \in \mathsf{T},\ \langle x - x', y - y'\rangle \geq \|y - y'\|^2.$$

The typical example of a monotone operator is the subdifferential ∂f of a convex function $f : \mathsf{Y} \to \mathbb{R}$. Finding a minimum of f amounts to finding a point in $\mathrm{zer}(\partial f)$, where $\mathrm{zer}(\mathsf{T}) = \{x \,:\, 0 \in \mathsf{T}(x)\}$ is the set of zeroes of an operator T. A common technique for finding a zero of a maximal monotone operator T is the so-called *proximal point algorithm* [15] that we now describe. The *resolvent* of T is the operator $\mathsf{J}_{\rho\mathsf{T}} \triangleq (\mathsf{I} + \rho\mathsf{T})^{-1}$ for $\rho > 0$. One key result (see *e.g.* [9]) says that T is maximal monotone if and only if $\mathsf{J}_{\rho\mathsf{T}}$ is firmly non expansive and its domain is Y. Observe that a firmly non expansive operator is single valued and denote by $\mathrm{fix}(\mathsf{J}_{\rho\mathsf{T}})$ the set of fixed points of $\mathsf{J}_{\rho\mathsf{T}}$. It is clear that $\mathrm{fix}(\mathsf{J}_{\rho\mathsf{T}}) = \mathrm{zer}(\mathsf{T})$. The firm non expansiveness of $\mathsf{J}_{\rho\mathsf{T}}$ plays a central role in the proof of the following result:

Lemma 2 (Proximal point algorithm [15]): If T is a maximal monotone operator and $\rho > 0$, then the iterates $\zeta^{k+1} = \mathsf{J}_{\rho\mathsf{T}}(\zeta^k)$ starting at any point of Y converge to a point of $\mathrm{fix}(\mathsf{J}_{\rho\mathsf{T}})$ whenever this set is non-empty.

B. Random Gauss-Seidel iterations

Assume now that the Euclidean space Y is a Cartesian product of Euclidean spaces of the form $\mathsf{Y} = \mathsf{Y}_1 \times \cdots \times \mathsf{Y}_L$ where L is a given integer, and write any $\zeta \in \mathsf{Y}$ as $\zeta = (\zeta_1, \ldots, \zeta_L)$ where $\zeta_\ell \in \mathsf{Y}_\ell$ for $\ell = 1, \ldots, L$. Let S be a firmly non expansive operator on Y and write

$$\mathsf{S}(\zeta) = (\mathsf{S}_1(\zeta), ..., \mathsf{S}_L(\zeta))$$

where $\mathsf{S}_\ell(\zeta) \in \mathsf{Y}_\ell$. For $\ell = 1, \ldots, L$, define the single valued operator $\hat{\mathsf{S}}_\ell : \mathsf{Y} \to \mathsf{Y}$ as

$$\hat{\mathsf{S}}_\ell(\zeta) = (\zeta_1, \ldots, \zeta_{\ell-1}, \mathsf{S}_\ell(\zeta), \zeta_{\ell+1}, \ldots, \zeta_L)\,. \tag{10}$$

Considering an iterative algorithm of the form $\zeta^{k+1} = \mathsf{S}(\zeta^k)$, its *Gauss-Seidel* version would be an algorithm of the form $\zeta^{k+1} = \hat{\mathsf{S}}_L \circ \cdots \circ \hat{\mathsf{S}}_1(\zeta^k)$. We are interested here in a *randomized version* of these iterates. On a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, let $(\xi^k)_{k\in\mathbb{N}}$ be a random process satisfying the following assumption:

Assumption 3: The random variables ξ^k are independent and identically distributed. They are valued in the set $\{1, \dots, L\}$ with $\mathbb{P}[\xi^1 = \ell] = p_\ell > 0$ for all $\ell = 1, \dots, L$.

We are interested here in the convergence of the random iterates $\zeta^{k+1} = \hat{\mathsf{S}}_{\xi^{k+1}}(\zeta^k)$ towards a (generally random) point of $\mathrm{fix}(\mathsf{S})$, provided this set is non empty:

Theorem 2 (Main result): Let S is a firmly non-expansive operator on Y with domain Y. Let $(\xi^k)_{k\in\mathbb{N}}$ be a sequence of random variables satisfying Assumption 3. Assume that $\mathrm{fix}(\mathsf{S}) \neq \emptyset$. Then for any initial value ζ^0, the sequence of iterates $\zeta^{k+1} = \hat{\mathsf{S}}_{\xi^{k+1}}(\zeta^k)$ converges almost surely to a random variable supported by $\mathrm{fix}(\mathsf{S})$.

Proof: Denote by $\langle \zeta, \eta \rangle = \sum_{\ell=1}^L \langle \zeta_\ell, \eta_\ell \rangle_{\mathsf{Y}_\ell}$ the inner product of Y, and by $\|\zeta\|^2 = \langle \zeta, \zeta \rangle$ its associated squared norm. Define a new inner product $\zeta \bullet \eta = \sum_{\ell=1}^K p_\ell^{-1} \langle \zeta_\ell, \eta_\ell \rangle_{\mathsf{Y}_\ell}$ on Y, and let $\|\!|\zeta|\!\|^2 = \zeta \bullet \zeta$ be its associated squared norm. Fix $\zeta^\star$ in $\mathrm{fix}(\mathsf{S})$. Conditionally to the sigma-field $\mathcal{F}_k = \sigma(\xi^1, \dots, \xi^k)$ we have

$$\begin{aligned}
\mathbb{E}[\|\!|\zeta^{k+1} - \zeta^\star|\!\|^2 \mid \mathcal{F}_k] &= \sum_{\ell=1}^L p_\ell \left\|\!\left| \hat{\mathsf{S}}_\ell(\zeta^k) - \zeta^\star \right|\!\right\|^2 \\
&= \sum_{\ell=1}^L p_\ell \Big(\frac{1}{p_\ell} \|\mathsf{S}_\ell(\zeta^k) - \zeta_\ell^\star\|_{\mathsf{Y}_\ell}^2 + \sum_{\substack{i=1 \\ i\neq\ell}}^L \frac{1}{p_i} \|\zeta_i^k - \zeta_i^\star\|_{\mathsf{Y}_i}^2 \Big) \\
&= \|\mathsf{S}(\zeta^k) - \zeta^\star\|^2 + \sum_{\ell=1}^L \frac{1-p_\ell}{p_\ell} \|\zeta_\ell^k - \zeta_\ell^\star\|_{\mathsf{Y}_\ell}^2 \\
&= \|\!|\zeta^k - \zeta^\star|\!\|^2 + \|\mathsf{S}(\zeta^k) - \zeta^\star\|^2 - \|\zeta^k - \zeta^\star\|^2
\end{aligned}$$

Since $(\mathsf{I} - \mathsf{S})(\zeta^\star) = 0$, we have

$$\begin{aligned}
&\|\mathsf{S}(\zeta^k) - \zeta^\star\|^2 - \|\zeta^k - \zeta^\star\|^2 \\
&= \|\mathsf{S}(\zeta^k) - \zeta^k + \zeta^k - \zeta^\star\|^2 - \|\zeta^k - \zeta^\star\|^2 \\
&= \|\mathsf{S}(\zeta^k) - \zeta^k\|^2 + 2\langle \mathsf{S}(\zeta^k) - \zeta^k, \zeta^k - \zeta^\star \rangle \\
&= \|\mathsf{S}(\zeta^k) - \zeta^k\|^2 - 2\langle (\mathsf{I} - \mathsf{S})(\zeta^k) - (\mathsf{I} - \mathsf{S})(\zeta^\star), \zeta^k - \zeta^\star \rangle \\
&\leq -\|\mathsf{S}(\zeta^k) - \zeta^k\|^2
\end{aligned}$$

where the inequality comes from the easily verifiable fact that $(\mathsf{I} - \mathsf{S})$ is firmly non-expansive when S is. This leads to the inequality

$$\mathbb{E}[\|\!|\zeta^{k+1} - \zeta^\star|\!\|^2 \,|\mathcal{F}_k] \leq \|\!|\zeta^k - \zeta^\star|\!\|^2 - \|\mathsf{S}(\zeta^k) - \zeta^k\|^2 \quad (11)$$

which shows that $\|\!|\zeta^k - \zeta^\star|\!\|^2$ is a nonnegative supermartingale with respect to the filtration $(\mathcal{F}_k)$. As such, it converges with probability one towards a random variable $X_{\zeta^\star}$ satisfying $0 \leq X_{\zeta^\star} < \infty$ almost everywhere. Given a countable dense subset H of $\mathrm{fix}(\mathsf{S})$, there is a probability one set on which $\|\!|\zeta^k - \boldsymbol{\zeta}|\!\| \to X_{\boldsymbol{\zeta}} \in [0, \infty)$ for all $\boldsymbol{\zeta} \in H$. Let $\zeta^\star \in \mathrm{fix}(\mathsf{S})$, let $\varepsilon > 0$, and choose $\boldsymbol{\zeta} \in H$ such that $\|\!|\zeta^\star - \boldsymbol{\zeta}|\!\| \leq \varepsilon$. With probability one, we have

$$\|\!|\zeta^k - \zeta^\star|\!\| \leq \|\!|\zeta^k - \boldsymbol{\zeta}|\!\| + \|\!|\boldsymbol{\zeta} - \zeta^\star|\!\| \leq X_{\boldsymbol{\zeta}} + 2\varepsilon$$

for k large enough. Similarly, $\|\!|\zeta^k - \zeta^\star|\!\| \geq X_{\boldsymbol{\zeta}} - 2\varepsilon$ for k large enough. We therefore obtain:

C1 : There is a probability one set on which $\|\!|\zeta^k - \zeta^\star|\!\|$ converges for every $\zeta^\star \in \mathrm{fix}(\mathsf{S})$.

Getting back to Inequality (11), taking the expectations on both sides of this inequality and iterating over k, we obtain

$$\sum_{k=0}^\infty \mathbb{E}[\|\mathsf{S}(\zeta^k) - \zeta^k\|^2] \leq (\zeta^0 - \zeta^\star)^2.$$

By Markov's inequality and Borel Cantelli's lemma, we therefore obtain:

C2 : $S(\zeta^k) - \zeta^k \to 0$ almost surely.

We now consider an elementary event in the probability one set where **C1** and **C2** hold. On this event, since $\|\!|\zeta^k - \zeta^\star|\!\|$ converges for $\zeta^\star \in \mathrm{fix}(\mathsf{S})$, the sequence ζ^k is bounded. Since S is firmly non expansive, it is continuous, and **C2** shows that all the accumulation points of ζ^k are in $\mathrm{fix}(\mathsf{S})$. It remains to show that these accumulation points reduce to one point. Assume that $\zeta_1^\star$ is an accumulation point. By **C1**, $\|\!|\zeta^k - \zeta_1^\star|\!\|$ converges. Therefore, $\lim \|\!|\zeta^k - \zeta_1^\star|\!\| = \liminf \|\!|\zeta^k - \zeta_1^\star|\!\| = 0$, which shows that $\zeta_1^\star$ is unique. ∎

V. Random ADMM

We now return to the optimization problem (2). It is a well known fact that the standard ADMM can be seen as special case of the so-called Douglas-Rachford algorithm [9]. The Douglas-Rachford algorithm can itself be seen as a special case of a proximal point algorithm. By the results of the previous section, this suggests that random Gauss-Seidel iterations applied to the Douglas-Rachford operator produce a sequence which eventually converges to the sought solutions. It turns out that the latter random iterations can be written under the form of practical asynchronous ADMM-like algorithm.

A. Douglas-Rachford operator

Consider the following dual problem associated with (2)

$$\min_{\lambda \in \mathsf{Z}} f^*(-M^*\lambda) + g^*(\lambda)\,, \quad (12)$$

where f^*, g^* are the Fenchel conjugates of f and g and M^* is the adjoint of M. By Assumption 1 along with [16, Th.3.3.5], the minimum in (12) is attained and its opposite coincides with the minimum of (2). Note that λ is a minimizer of (12) iff zero belongs to the subdifferential of the objective function in (12). By [16, Th.3.3.5] again, this reads $0 \in -M \cdot \partial f^*(-M^*\lambda) + \partial g^*(\lambda)$. Otherwise stated, finding minimizers of the dual problem (12) boils down to searching zeros of the sum of two maximal monotone operators $\mathsf{T} + \mathsf{U}$ defined by $\mathsf{T} = -M \cdot \partial f^* \circ (-M^*)$ and $\mathsf{U} = \partial g^*$. For a fixed $\rho > 0$, the Douglas-Rachford / Lions-Mercier operator R is defined as

$$\{(\nu + \rho b, \mu - \nu) : (\mu, b) \in \mathsf{U}, (\nu, a) \in \mathsf{T}, \nu + \rho a = \mu - \rho b\}\,.$$

The following Lemma is an immediate consequence of [9].

Lemma 3: Under Assumption 1, R is maximal monotone, and $\mathrm{zer}(\mathsf{R}) \neq \emptyset$. Moreover, $\mathsf{J}_{\rho\mathsf{U}}(\zeta) \in \mathrm{zer}(\mathsf{T}+\mathsf{U})$ for any $\zeta \in \mathrm{zer}(\mathsf{R})$.

Lemma 3 implies that the search for a zero of $\mathsf{T}+\mathsf{U}$ boils down to the search of a zero of R up to a resolvent step $\mathsf{J}_{\rho\mathsf{U}}$. To that end, a standard approach is to use a proximal point algorithm of the form $\zeta^{k+1} = \mathsf{J}_{\mathsf{R}}(\zeta^k)$. By [9], it can be shown that this approach is equivalent to the ADMM derived in Section II. Here, our aim is different. We shall consider random Gauss-Seidel iterations in order to derive an asynchronous version of the ADMM.

B. Random Gauss-Seidel Iterations

Define $\mathsf{S} \triangleq \mathsf{J}_{\mathsf{R}}$ as the resolvent associated with the Douglas-Rachford operator R. On the space $\mathsf{Z} = \mathsf{X}^{A_1} \times \cdots \times \mathsf{X}^{A_L}$, define the operator $\hat{\mathsf{S}}_\ell$ as in (10) for any $\ell = 1, \cdots, L$. Let $(\xi^k)_{k\in\mathbb{N}}$ be a random process satisfying Assumption 3. The following result is a consequence of Theorem 2 combined with Lemma 3.

Theorem 3: Let Assumptions 1, 2 and 3 hold true. Consider the sequence $(\zeta^k)_k$ defined by $\zeta^{k+1} = \hat{\mathsf{S}}_{\xi^{k+1}}(\zeta^k)$. Then for any initial value ζ^0, the sequence $\lambda^k \triangleq \mathsf{J}_{\rho\mathsf{U}}(\zeta^k)$ converges almost surely to a minimizer of (12).

In order to complete the above result, we still must justify the fact that, as claimed, the above iterations can be seen as an asynchronous distributed algorithm.

C. Distributed Algorithm

We make the above random Gauss-Seidel iterations more explicit. In the sequel we shall always denote by ζ_ℓ the ℓth component of a function $\zeta \in \mathsf{Z}$ *i.e.*, $\zeta = (\zeta_1, \cdots, \zeta_L)$. For any ℓ, we introduce the average $\bar{\zeta}_\ell = \sum_{v\in A_\ell} \zeta_\ell(v)/|A_\ell|$. Lemma 4 below states that any $\zeta \in \mathsf{Z}$ is uniquely represented by a couple $(\lambda, z) \in \mathsf{U}$ whose expression is provided. Moreover, it provides the explicit form of the ℓth block S_ℓ of the resolvent S. This shall be the basis of our asynchronous distributed algorithm.

Lemma 4: For any $\zeta \in \mathsf{Z}$, the following holds true.

i) There exist a unique $(\lambda, z) \in \mathsf{U}$ such that $\lambda + \rho z = \zeta$.

ii) $\mathsf{J}_{\rho U}(\zeta) = \lambda$.

iii) For any $\ell = 1, \cdots, L$, $\lambda_\ell = \zeta_\ell - \bar{\zeta}_\ell 1_{A_\ell}$ and $z_\ell = \frac{\bar{\zeta}_\ell}{\rho} 1_{A_\ell}$.

iv) For any $\ell = 1, \cdots, L$, and any $v \in A_\ell$

$$\mathsf{S}_\ell(\zeta) : v \mapsto \lambda_\ell(v) + \rho x(v) \tag{13}$$

where $x(v)$ is defined by

$$x(v) = \mathrm{prox}_{f_v, \rho|\sigma(v)|}\left(\frac{1}{|\sigma(v)|} \sum_{\ell\in\sigma(v)} \bar{z}_\ell - \frac{\lambda_\ell(v)}{\rho} \right). \tag{14}$$

Proof: i)-ii) Existence: Let us define $\lambda = \mathsf{J}_{\rho\mathsf{U}}(\zeta)$ and $z = (\zeta - \lambda)/\rho$. Trivially, $\lambda + \rho z = \zeta$. As $\zeta \in \lambda + \rho\mathsf{U}(\lambda)$, we deduce that $(\lambda, z) \in \mathsf{U}$. *Uniqueness*: For a fixed $(\lambda, z) \in \mathsf{U}$ satisfying $\lambda + \rho z = \zeta$, one has $\zeta \in (I + \rho\mathsf{U})(\lambda)$ and thus $\lambda = \mathsf{J}_{\rho\mathsf{U}}(\zeta)$. As a consequence, $z = (\zeta - \lambda)/\rho$.

iii) We use $\lambda = \mathsf{J}_{\rho\mathsf{U}}(\zeta) = \mathrm{prox}_{g^*,\rho}(\zeta) = \zeta - \mathrm{prox}_{g,\rho}(\zeta)$ (see [17, Th. 14.3]). As g is the indicator function of the set $\mathrm{sp}(1_{A_1}) \times \cdots \times \mathrm{sp}(1_{A_L})$, $\mathrm{prox}_{g,\rho}$ coincides with the projection operator onto that set. Thus, for any ℓ, $\lambda_\ell = \zeta_\ell - \bar{\zeta}_\ell 1_{A_\ell}$. The expression of z follows from $z = (\zeta - \lambda)/\rho$.

iv) Operator $\mathsf{S} = \mathsf{J}_{\mathsf{R}}$ can be written as

$$\{(\mu + \rho b, \nu + \rho b) : (\mu, b) \in \mathsf{U}, (\nu, a) \in \mathsf{T}, \nu + \rho a = \mu - \rho b\}.$$

Moreover, as R is monotone, $\mathsf{S}(\zeta)$ is a singleton. Representing $\zeta = \lambda + \rho z$ with $(\lambda, z) \in \mathsf{U}$, it follows from the above expression of S that $\mathsf{S}(\zeta) = \nu + \rho z$ where ν is such that $\nu + \rho a = \lambda - \rho z$ for some $a \in \mathsf{T}(\nu)$. Using $\mathsf{T} = -M \cdot \partial f^* \circ (-M^*)$, condition $a \in \mathsf{T}(\nu)$ translates to: there exists $x \in \partial f^*(M^*\nu)$ s.t. $a = -Mx$. The output-resolvent is obtained by $\nu + \rho z = \lambda + \rho M x$. For a given component ℓ, this boils down to equation (13). The remaining task is to provide the expression of x. By the Fenchel-Young equality $\partial f^* = \partial f^{-1}$ [16, Prop.3.3.4], condition $x \in \partial f^*(M^*\nu)$ is equivalent to $M^*\nu \in \partial f(x)$. Using that $\nu = \lambda - \rho(z - Mx)$, we obtain $0 \in \partial f(x) - M^*\lambda + \rho M^*(z - Mx)$. Otherwise stated, $x = \arg\min_{y\in\mathsf{X}^V} \mathcal{L}_\rho(y, z; \lambda)$ where $\mathcal{L}_\rho$ is the augmented Lagrangian defined in (3). Using the results of Section III, $x(v)$ is given by (14) for any v. ∎

We are now in position to state the main algorithm. It simply consists in an explicit writing of the random Gauss-Seidel iterations $\zeta^{k+1} = \hat{\mathsf{S}}_{\xi^{k+1}}(\zeta^k)$ using Lemma 4*iv)*. Note that, by Lemma 4*i)*, the definition of a sequence $(\zeta_k)_k$ on Z is equivalent to the definition of two sequences $(\lambda^k, z^k) \in \mathsf{U}$ such that $\zeta^k = \lambda^k + \rho z^k$. Moreover, by Lemma 4*iii)*, each component z_ℓ^k of z^k is a constant. The definition of z^k thus reduces to the definition of L constants $\bar{z}_1^k, \cdots, \bar{z}_L^k$ in X.

Asynchronous ADMM:

At each iteration k, draw r.v. ξ^{k+1}.

For $\ell = \xi^{k+1}$, set for any $v \in A_\ell$:

$$\begin{aligned}
x^{k+1}(v) &= \mathrm{prox}_{f_v, \rho|\sigma(v)|}\left(\frac{1}{|\sigma(v)|} \sum_{\ell\in\sigma(v)} \bar{z}_\ell^k - \frac{\lambda_\ell^k(v)}{\rho} \right) \\
\bar{z}_\ell^{k+1} &= \frac{1}{|A_\ell|} \sum_{w\in A_\ell} x^{k+1}(w) \\
\lambda_\ell^{k+1}(v) &= \lambda^k(v) + \rho\left(x^{k+1}(v) - \bar{z}_\ell^{k+1} \right).
\end{aligned}$$

For any $\ell \neq \xi^{k+1}$, set $\lambda_\ell^{k+1} = \lambda_\ell^k$.

For any $w \notin A_{\xi^{k+1}}$, set $x^{k+1}(w) = x^k(w)$.

VI. Implementation Example

In order to illustrate our results, we consider herein an asynchronous version of the ADMM algorithm in the context of Section II-Example 2. The scenario is the following: first, Agent $v \in \{1, \ldots, |V|\}$ wakes up at time $k+1$ with the probability q_v. Denoting by $\mathcal{N}_v$ the neighborhood of Agent v in the Graph G, this agent then chooses one of its neighbors, say w, with the probability $1/|\mathcal{N}_v|$ and sends an activation message to w. In this setting, the edge $\{v, w\}$ coincides with one of the A_ℓ of Example 2 in Section II. It is easy to see that the samples of the activation process ξ^k who is of course valued in E are governed by the probability law

$$\mathbb{P}[\xi^1 = \{v, w\}] = \frac{q_v}{|\mathcal{N}_v|} + \frac{q_w}{|\mathcal{N}_w|} > 0.$$

When the edge $\{v,w\}$ is activated, the following two prox$(\cdot)$ operations are performed by the agents:

$$x^{k+1}(v) = \text{prox}_{f_v,\rho|\mathcal{N}(v)|}\Big(\frac{1}{|\mathcal{N}(v)|}\sum_{\ell\in\mathcal{N}(v)}\bar{z}_\ell^k - \frac{\lambda_\ell^k(v)}{\rho}\Big)$$

$$x^{k+1}(w) = \text{prox}_{f_w,\rho|\mathcal{N}(w)|}\Big(\frac{1}{|\mathcal{N}(w)|}\sum_{\ell\in\mathcal{N}(w)}\bar{z}_\ell^k - \frac{\lambda_\ell^k(w)}{\rho}\Big).$$

The two agents exchange then the values $x^{k+1}(v)$ and $x^{k+1}(w)$ and perform the following operations:

$$\bar{z}_\ell^{k+1} = \frac{x^{k+1}(v)+x^{k+1}(w)}{2}$$

$$\lambda_\ell^{k+1}(v) = \lambda^k(v)+\rho\frac{x^{k+1}(v)-x^{k+1}(w)}{2}$$

$$\lambda_\ell^{k+1}(w) = \lambda^k(w)+\rho\frac{x^{k+1}(w)-x^{k+1}(v)}{2}.$$

We remark that this communication scheme is reminiscent of the so-called *Random Gossip* algorithm introduced in [18] in the context of distributed averaging.

VII. Numerical Results

We consider a network with $V=\{1,\dots,5\}$ and with $E=\{\{1,2\},\{2,3\},\{3,4\},\{4,5\},\{5,3\}\}$. We evaluate the behavior of: i) the *Synchronous ADMM* ii) the *Asynchronous ADMM* and iii) the *Distributed Gradient Descent* with $1/\sqrt{k}$ stepsize [19] using *Random Gossip* as a communication algorithm[18]. Each agent maintains a different quadratic convex function and their goal is to reach consensus over the minimizer of problem (1).

In Figure 1, we plot the squared error versus the number of primal updates for the three considered algorithms. We observe that our algorithm clearly outperforms the Distributed Gradient Descent.

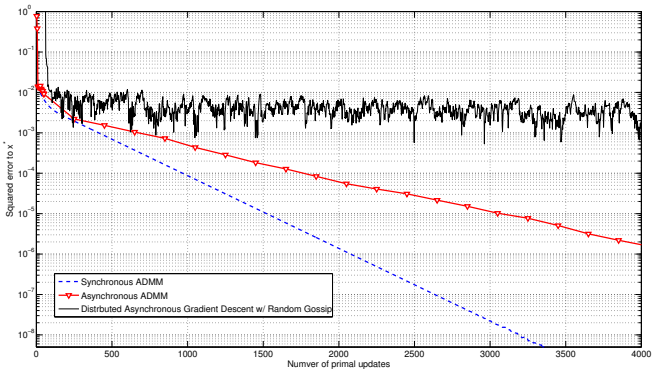

Fig. 1. Squared error of distributed optimization algorithms versus the number of primal updates

References

[1] P. A. Forero, A. Cano, and G. B. Giannakis, "Distributed Clustering Using Wireless Sensor Networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 707–724, 2011.

[2] Dimitri P Bertsekas and John N Tsitsiklis, *Parallel and distributed computation*, Old Tappan, NJ (USA); Prentice Hall Inc., 1989.

[3] S. Ram, A. Nedic, and V. Veeravalli, "Distributed Stochastic Subgradient Projection Algorithms forConvex Optimization," *Journal of Optimization Theory and Applications*, vol. 147, no. 3, pp. 516–545, 2010.

[4] P. Bianchi and J. Jakubowicz, "Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization," *IEEE Transactions on Automatic Control*, vol. 58, no. 2, 2013.

[5] Dusan Jakovetić, Jos M. F. Moura, and Xavier Joao, "Distributed Nesterov-like gradient algorithms," in *Proc. 51st IEEE Conference on Decision and Control (CDC)*, 2012, pp. 5459–5464.

[6] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual Averaging for Distributed Optimization: Convergence Analysis and Network Scaling," *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 592–606, 2012.

[7] Ali Jadbabaie, Asuman Ozdaglar, and Michael Zargham, "A distributed newton method for network optimization," in *Proc. 48th IEEE Conference on Decision and Control (CDC)*, 2009, pp. 2736–2741.

[8] P.L. Lions and B. Mercier, "Splitting algorithms for the sum of two nonlinear operators," *SIAM Journal on Numerical Analysis*, vol. 16, no. 6, pp. 964–979, 1979.

[9] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, pp. 293–318, 1992.

[10] Patrick L. Combettes and Jean-Christophe Pesquet, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, chapter Proximal Splitting Methods in Signal Processing, pp. 185–222, Springer, 2011.

[11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, vol. 3 of *Foundations and Trends in Machine Learning*, Now Publishers Inc., 2011.

[12] I.D. Schizas, A. Ribeiro, and G.B. Giannakis, "Consensus in Ad Hoc WSNs With Noisy Links – Part I: Distributed Estimation of Deterministic Signals," *Signal Processing, IEEE Transactions on*, vol. 56, no. 1, pp. 350 –364, jan. 2008.

[13] Joao F. C. Mota, Joao M. F. Xavier, Pedro M. Q. Aguiar, and Markus Puschel, "Distributed ADMM for model predictive control and congestion control," in *Proc. 51st IEEE Conference on Decision and Control (CDC)*, 2012, pp. 5110–5115.

[14] Ermin Wei and Asuman Ozdaglar, "Distributed Alternating Direction Method of Multipliers," in *Proc. 51st IEEE Conference on Decision and Control (CDC)*, 2012, pp. 5445–5450.

[15] R Tyrrell Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM Journal on Control and Optimization*, vol. 14, no. 5, pp. 877–898, 1976.

[16] J.M. Borwein and A.S. Lewis, *Convex Analysis and Nonlinear Optimization : Theory and Examples*, Springer Verlag, 2006.

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

[18] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

[19] John N. Tsitsiklis, Dimitri P. Bertsekas, and Mlichael Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.