Distributed Projection on the Simplex and ℓ_1 Ball via ADMM and Gossip

Franck Iutzeler and Laurent Condat

***Abstract*—We derive distributed algorithms for projecting the local values of the agents of a computing network on the simplex or on the ℓ_1 ball. These algorithms are based on the distributed alternating direction method of multipliers to solve a convex optimization problem of the form $\min_x \sum_n f_n(x)$, where each function f_n is local to node n and has an easy-to-compute proximity operator.**

***Index Terms*—Distributed optimization, gossiping, l1 norm ball, network, projection, simplex.**

I. Introduction

Let us consider a connected network of N agents, each with a scalar value a_n. The problem of averaging, learning, projecting, or, broadly speaking, minimizing a function of these values $(a_n)_{n=1,\dots,N}$ in a distributed fashion has been receiving a lot of attention [1]–[4]. Indeed, whenever computing or data storing resources are distributed, these agents have to exchange in order to attain a common goal. These communications often have to happen in a local way, one agent to another (instead of relying on broadcasts with a master node), for robustness (in computer networks) or due to reachability (in Wireless Sensor Networks). Algorithms that distributively solve problems using local, possibly asynchronous, communications are usually called *gossip* algorithms.

Since the seminal work [4] on gossip for averaging, many works investigated gossip algorithms for averaging and more involved problems such as maximal value estimation, or first-order optimization [4]–[13]. For solving more general optimization problems, distributed synchronous [14], [15] and asynchronous [16], [17] algorithms were proposed. The general idea of these methods is to formulate a distributed optimization problem (as the sum of local losses plus a consensus enforcing term), and solving it using (synchronous or asynchronous) splitting-based optimization methods such as the popular Alternating Direction Method of Multipliers (ADMM) [14], [16], [18]–[20], or Primal-Dual algorithms [21]–[23].

Recently, the first author proposed a distributed algorithm for quantile or median computation [24] based on

i) reformulating the problem into the minimization of $F(x) = \sum_n f_{a_n}(x)$ where each f_{a_n} is an $\mathbb{R} \to \mathbb{R}$, convex, function, with an easy-to-compute proximity operator, depending only on the data point a_n;

ii) building on the structure F as a sum of local prox-easy function to use distributed ADMM [14], [15], [25] to generate distributed algorithms solving the original problem.

In this letter, we address the problem of projecting the agent values onto the simplex and the ℓ_1 ball in a distributed way. Efficiently handling these projections is important for signal processing applications as there are at the heart of convex relaxations of nonconvex labeling problems, as well as of unmixing problems, see [26], [27] and references therein. The approach has the same flavor as the one proposed in [24], but there are significant differences between the problems of computing averages or quantiles and projecting onto the aforementioned sets. Therefore, the proposed formulation as an optimization problem is completely novel.

First, after recalling known results about simplex and ℓ_1-ball projection, we introduce an auxiliary problem in the form of a sum of local prox-easy functions in Section III. Then, in Section IV, we rely on distributed ADMM to derive the actual distributed algorithm for projecting onto the target sets. Finally, we provide numerical illustrations of our algorithms in Section V.

II. Projecting on the Simplex and ℓ_1-Ball

First, let us briefly recall some definitions and results (see [28] for a detailed overview). Let $\mathbf{a} = (a_1, \dots, a_N) \in \mathbb{R}^N$ and let $\|\cdot\|$ be the Euclidean norm. Out of convenience, we denote by $\mathbf{a} + t$ the addition of the scalar t to each coordinate of the vector $\mathbf{a}$. The projections of a onto the simplex and onto the ℓ_1-ball are respectively defined as

$$\mathcal{P}_\Delta(\mathbf{a}) = \operatorname*{arg\,min}_{\mathbf{y}\in\Delta} \|\mathbf{y} - \mathbf{a}\| \quad \text{and}$$

$$\mathcal{P}_{\mathsf{B}}(\mathbf{a}) = \operatorname*{arg\,min}_{\mathbf{y}\in\mathsf{B}} \|\mathbf{y} - \mathbf{a}\|,$$

where Δ is the *simplex* defined as the set of nonnegative vectors whose elements sum up to some fixed constant $\mathsf{s} > 0$ (the usual

Manuscript received June 6, 2018; revised August 30, 2018; accepted September 13, 2018. Date of publication September 17, 2018; date of current version September 26, 2018. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Augusto Aubry. *(Corresponding author: Franck Iutzeler.)*

F. Iutzeler is with Univ. Grenoble Alpes, Laboratoire Jean Kuntzmann, CNRS 38000 Grenoble, France (e-mail: franck.iutzeler@univ-grenoble-alpes.fr).

L. Condat is with Univ. Grenoble Alpes, GIPSA-Lab, CNRS 38000 Grenoble, France (e-mail: laurent.condat@gipsa-lab.grenoble-inp.fr).

This letter has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2018.2870951

probability or unit simplex corresponds to $\mathsf{s} = 1$); that is,

$$\Delta = \left\{ \mathbf{y} = (y_1, \ldots, y_N) \in \mathbb{R}_+^N : \sum_{n=1}^{N} y_n = \mathsf{s} \right\} \tag{1}$$

and B is the ℓ_1-*ball*:

$$\mathsf{B} = \left\{ \mathbf{y} = (y_1, \ldots, y_N) \in \mathbb{R}^N : \sum_{n=1}^{N} |y_n| \le \mathsf{s} \right\}. \tag{2}$$

These two projections are closely related; in fact, they can be obtained one from the other as detailed in the following lemma:

Lemma 1 ([29, Lemma 3]):

$$\mathcal{P}_{\mathsf{B}}(\mathbf{a}) = \begin{cases} \mathbf{a} & \text{if } \sum_{n=1}^{N} |a_n| \le \mathsf{s}, \\ \operatorname{sign}(\mathbf{a}) \odot \mathcal{P}_\Delta(|\mathbf{a}|) & \text{else,} \end{cases} \tag{3}$$

$$\mathcal{P}_\Delta(\mathbf{a}) = \mathcal{P}_{\mathsf{B}}(\mathbf{a} - \min_n a_n + \mathsf{s}/N), \tag{4}$$

where sign is the sign function ($[\operatorname{sign}(\mathbf{x})]_n = 1$ if $x_n > 0$, -1 if $x_n < 0$, and 0 else), $\odot$ denotes the elementwise product, and $|\mathbf{a}|$ is the vector of the absolute values of the elements of a.

Consequently, we focus in the following on the projection onto the simplex; this projection amounts to finding a suitable threshold value, in the sense of the following result:

Lemma 2 ([30]): There is a unique $t \in \mathbb{R}$ such that

$$\mathcal{P}_\Delta(\mathbf{a}) = [\mathbf{a} - t]_+, \tag{5}$$

where $[\cdot]_+$ is the elementwise maximum with 0.

Since $\sum_{n=1}^{N} [\mathcal{P}_\Delta(\mathbf{a})]_n = \mathsf{s}$, one can directly see that the problem of projecting onto the simplex boils down to finding a threshold t satisfying

$$\sum_{n: a_n > t} (a_n - t) = \mathsf{s}, \tag{6}$$

which is the actual problem that we will solve in a distributed manner.

III. The Projection Onto the Simplex as an Optimization Problem

In this section, we formulate an optimization problem, based on proposed *simplex objective functions*, the solution of which satisfies the equality of (6).

A. Objective Functions

We define the *simplex objective function* (f_α), for any $\alpha \in \mathbb{R}$, as

$$f_\alpha(t) = \begin{cases} \dfrac{1}{2}(t - \alpha)^2 + \dfrac{\mathsf{s}}{N} t & \text{if } t \le \alpha, \\ \dfrac{\mathsf{s}}{N} t & \text{else.} \end{cases} \tag{7}$$

It is straightforward to see that for any $\alpha \in \mathbb{R}$, f_α is a $\mathcal{C}^1$ convex function. In addition, its proximity operator can be explicitly computed for any $\gamma > 0$ and any $z \in \mathbb{R}$ as

$$\begin{aligned} \mathbf{prox}_{\gamma f_\alpha}(z) &\triangleq \operatorname*{arg\,min}_{t \in \mathbb{R}} \left\{ f_\alpha(t) + \frac{1}{2\gamma} \|t - z\|^2 \right\}, \\ &= \begin{cases} \dfrac{z + \gamma\left(a - \dfrac{\mathsf{s}}{N}\right)}{1 + \gamma} & \text{if } z \le \alpha + \gamma \dfrac{\mathsf{s}}{N}, \\ z - \gamma \dfrac{\mathsf{s}}{N} & \text{else.} \end{cases} \end{aligned} \tag{8}$$

Indeed, set $\bar{t} = \mathbf{prox}_{\gamma f_\alpha}(z)$. If $\bar{t} \le \alpha$, we have, from the first order optimality conditions, $\bar{t} - \alpha + \mathsf{s}/N + (\bar{t} - z)/\gamma = 0$, thus $\bar{t} = (z + \gamma(\alpha - \mathsf{s}/N))/(1 + \gamma)$; moreover, the condition $\bar{t} \le \alpha$ translates on the input z as $z \le \alpha + \gamma \mathsf{s}/N$. Similarly, when $\bar{t} > \alpha$, one gets $\mathsf{s}/N + (\bar{t} - z)/\gamma = 0$, which leads to the second part of the result.

B. Equivalent Problem

The previously defined functions enable us to formulate our problem of interest for an input vector $a \in \mathbb{R}^N$ as

$$\min_{t \in \mathbb{R}} f(t) \triangleq \sum_{n=1}^{N} f_{a_n}(t). \tag{9}$$

Theorem 1: Problem (9) has a unique minimizer $t^\star$ and

$$\mathcal{P}_\Delta(\mathbf{a}) = [\mathbf{a} - t^\star]_+, \tag{10}$$

with f locally μ-strongly convex around $t^\star$ and $\mu = \operatorname{card}(\{n : a_n > t^\star\}) \ge 1$.

Proof: As (9) is an unconstrained minimization of a smooth function, we have from the first-order optimality conditions that:

$$\begin{aligned} 0 = \nabla f(t^\star) &= \sum_{n: a_n > t^\star} (t^\star - a_n + \mathsf{s}/N) + \sum_{n: a_n \le t^\star} \mathsf{s}/N \\ &= \sum_{n: a_n > t^\star} (t^\star - a_n) + \mathsf{s}. \end{aligned} \tag{11}$$

Thus, $\sum_{n: a_n > t^\star} (a_n - t^\star) = \mathsf{s}$ which matches the sought-after relation of (6) and thus $\mathcal{P}_\Delta(\mathbf{a}) = [\mathbf{a} - t^\star]_+$ from Lemma 2. Finally, set $\bar{a} = \min(\{a_n : a_n > t^\star\})$ and $\mu = \operatorname{card}(\{n : a_n > t^\star\}) \ge 1$ (as $\sum_{n=1}^{N} \mathcal{P}_\Delta(\mathbf{a})_n = \sum_{n=1}^{N} [a_n - t^\star]_+ = \mathsf{s} > 0$, there is necessarily at least one a_n such that $a_n > t^\star$). For every $t \in (-\infty, \bar{a}]$, we have

$$\begin{aligned} f(t) \triangleq \sum_{n=1}^{N} f_{a_n}(t) &= \sum_{n: a_n > t^\star} f_{a_n}(t) + \sum_{n: a_n \le t^\star} f_{a_n}(t) \\ &= \sum_{n: a_n > t^\star} \left(\frac{1}{2}(t - a_n)^2 + \frac{\mathsf{s}}{N} t \right) + \sum_{n: a_n \le t^\star} f_{a_n}(t) \\ &= \frac{\operatorname{card}(\{n : a_n > t^\star\})}{2} t^2 \\ &\quad + \underbrace{\sum_{n: a_n > t^\star} \left(\frac{1}{2} a_n^2 + \left(\frac{\mathsf{s}}{N} - a_n \right) t \right) + \sum_{n: a_n \le t^\star} f_{a_n}(t)}_{\text{convex}} \end{aligned}$$

as if $a_n > t^\star$, then, for $t \le \bar{a} = \min(\{a_n : a_n > t^\star\})$, $f_{a_n}(t) = \frac{1}{2}(t - a_n)^2 + (\mathsf{s}/N)t$. The part in braces is convex, as a sum of

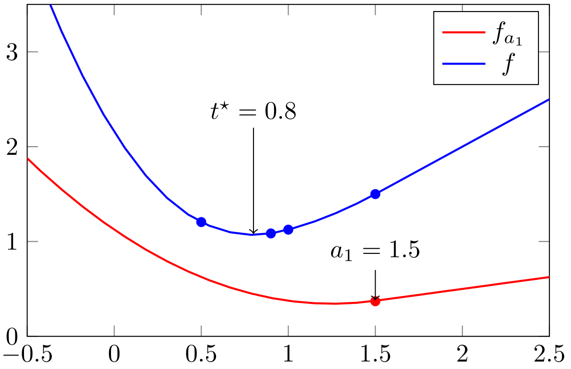


Fig. 1. Illustration of f_{a_1} and f for $\mathbf{a} = [1.5, 0.5, 1.0, 0.9]$ and $\mathsf{s} = 1$, leading to $t^\star = 0.8$ and $\mathcal{P}_\Delta(\mathbf{a}) = [0.7, 0.0, 0.2, 0.1]$. The dots on the graph represent the junctions between the parts of the curve. One can see that the function is locally 3-strongly convex on the interval $[0.5, 0.9]$ around $t^\star$.

convex functions, thus $f(t) - (\mu/2)t^2$ is convex on $(-\infty, \overline{a}]$. Hence f is μ-strongly convex on $(-\infty, \overline{a}]$, and since $t^\star$ lies in the interior of this interval, one can say that f is μ-strongly convex around $t^\star$, which yields the uniqueness of $t^\star$. ■

Fig. 1 provides a representation of a simplex objective function and the global objective function f, in a simple example.

Finally, let us notice that although for any α, the function f_α is 1-smooth (differentiable with 1-Lipschitz gradient), it has a linear part on which the gradient is constant and thus gradient descent may be arbitrarily slow depending on the initialization. This naturally leads us to consider proximal minimization algorithms like ADMM or primal-dual algorithms (see [31, Chap. 27]).

IV. Distributed Projection on the Simplex

Consider a network of N agents, the agent number n knowing only the data value $a_n \in \mathbb{R}$. The connected, undirected network is represented by its set of edges $E = \{(n,m) :$ the agents n and m are connected$\}$. The agent number n can send and receive information from its neighborhood $\mathcal{N}_n = \{m \neq n : (n,m) \in E\}$. We denote by $d_n = \mathrm{card}(\mathcal{N}_n)$ its degree (number of neighbors).

A. Proposed Algorithm

The objective of this section is to design an algorithm that distributively projects the full vector $\mathbf{a}$ onto the simplex. To do so, we first reformulate our problem in a distributed way and use ADMM to solve it; this results in an algorithm using the proximity operators of the $(f_{a_n})_n$ and local exchanges between the agents (see [6], [24], [25] for details about the link between ADMM and gossip-based optimization).

In order to formulate Problem (9) in a distributed manner, one can introduce a vector $\mathbf{x} \in \mathbb{R}^N$ and associate each coordinate n with an agent n, a data point a_n, and the corresponding simplex objective function f_{a_n}. Problem (9) is equivalent to minimizing $\sum_{n=1}^N f_{a_n}(x_n)$ under the constraint that $x_1 = x_2 = \cdots = x_N$ in the sense that the solution of the latter is of the form $(t^\star, .., t^\star)$ where $t^\star$ is the solution of the former. Finally, to take into account the links between the agents, the constraint $x_1 = x_2 = \cdots = x_N$ can be reformulated as $x_i = x_j$ for all $(i,j) \in E$ as the graph is connected; putting the constraints in the functions, one gets the distributed problem

$$\min_{\mathbf{x} \in \mathbb{R}^N} \sum_{n=1}^{N} f_{a_n}(x_n) + \sum_{n=1^N} c_n(\mathbf{x}), \tag{12}$$

where $c_n(\mathbf{x}) = 0$ if $x_n = x_j$ for all $j \in \mathcal{N}_n$ and $+\infty$ elsewhere. Once again, the solution of (12) is of the form $(t^\star, .., t^\star)$ where $t^\star$ is the solution of (9). Applying ADMM on it (see the monograph [14] and references therein), leads to the following algorithm after simplification of some variables omitted here due to lack of space. This methods relies on an hyper-parameter ρ that can be chosen as any positive real value (although it might affect the convergence rate), a typical choice which seems to be practically sound is $\rho = 1$.

We can note that the overall memory usage is $O(|E|)$, and the computational complexity per iteration is $O(1)$. Finally, note that step (13b) may not be computed at each iteration as it gives the sought value from an algorithm variable.

Distributed Projection on the Simplex

Initialization: $\mathbf{z}^0, \lambda^0 \in \mathbb{R}^N$, $\rho > 0$.

At each iteration $k = 0, 1, \ldots$

▷ [Computation] for each agent n

$$x_n^{k+1} = \begin{cases} \dfrac{\rho d_n z_n^k + a_n - \mathsf{s}/N}{1 + \rho d_n} & \text{if } z_n^k \leq a_n + \frac{\mathsf{s}}{\rho d_n n}, \\ z_n^k - \dfrac{\mathsf{s}}{\rho d_n n} & \text{else,} \end{cases} \tag{13a}$$

$$p_n^{k+1} = \max(a_n - x_n^{k+1}, 0). \tag{13b}$$

▷ [Exchange] all agents send their versions of x^{k+1} to their neighbors

$$\bar{x}_n^{k+1} = \frac{1}{d_n} \sum_{m \in \mathcal{N}_n} x_m^{k+1},\ \lambda_n^{k+1} = \lambda_n^k + \frac{x_n^{k+1} - \bar{x}_n^{k+1}}{2},$$

$$z_n^{k+1} = \frac{x_n^{k+1} + \bar{x}_n^{k+1}}{2} - \lambda_n^{k+1}. \tag{14}$$

Theorem 2: The *Distributed Projection on the Simplex* produces a sequence $(\mathbf{x}^k)$ that converges linearly to a consensus over a value $t^\star$ satisfying $\mathcal{P}_\Delta(\mathbf{a}) = [\mathbf{a} - t^\star]_+$, and

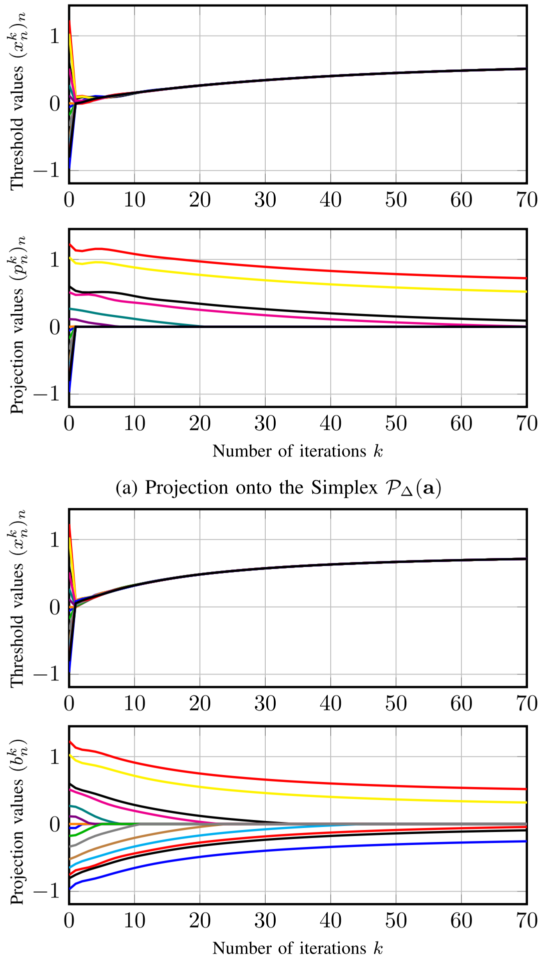
$$\mathbf{p}^k \to \mathcal{P}_\Delta(\mathbf{a}) \quad \text{as } k \to \infty. \tag{15}$$

Proof: This theorem comes from the fact that the *Distributed Projection on the Simplex* is exactly ADMM applied to the equivalent problem (12). Since f is locally strongly convex around its optimum $t^\star$, $(\mathbf{x}^k)$ converges linearly to $t^\star$ (see [25, Th. 2] for the proof and the explicit rate depending on the graph). Finally, Theorem 1 allows us to conclude that (15) holds. ■

Remark 1: This algorithm can be extended to asynchronous versions where only some randomly drawn agents compute and update at each iterations by applying the same reasoning as in [16], [17], [23], [24].

B. Distributed Projection onto the ℓ_1 Ball

Following the results of Section II, it is straightforward to adapt the previous algorithm in order to project on the ℓ_1 ball



(a) Projection onto the Simplex $\mathcal{P}_\Delta(\mathbf{a})$

(b) Projection onto the ℓ_1-ball $\mathcal{P}_{\mathsf{B}}(\mathbf{a})$

Fig. 2. Evolution of the variables of the algorithm with respect to the number of iterations. Each color corresponds to one agent n. (a) Projection onto the Simplex $\mathcal{P}_\Delta(\mathbf{a})$. (b) Projection onto the ℓ_1-ball $\mathcal{P}_{\mathsf{B}}(\mathbf{a})$.

rather than on the simplex. Indeed, instead of applying the *Distributed Projection on the Simplex* on $\mathbf{a}$, it suffices to apply it on $|\mathbf{a}|$. Then, the rule of (3) can be applied to deduce the projection of $\mathbf{a}$ on the ℓ_1-ball from the projection of $|\mathbf{a}|$ on the simplex. In practice, it suffices to replace (13a-13b) by

$$x_n^{k+1} = \begin{cases} \dfrac{\rho d_n z_n^k + |a_n| - \mathsf{s}/N}{1+\rho d_n} & \text{if } z_n^k \le |a_n| + \dfrac{\mathsf{s}}{\rho d_n n}, \\[2ex] z_n^k - \dfrac{\mathsf{s}}{\rho d_n n} & \text{else,} \end{cases} \tag{16a}$$

$$b_n^{k+1} = \begin{cases} \operatorname{sign}(a_n)\max(|a_n| - x_n^{k+1}, 0) & \text{if } x_n^{k+1} \ge 0, \\ a_n & \text{else.} \end{cases} \tag{16b}$$

Then the vector $\mathbf{b}^k$ converges to $\mathcal{P}_{\mathsf{B}}(\mathbf{a})$.

V. Numerical Experiments

In this section, we provide an illustration of the features of our algorithms on an undirected graph of $N = 15$ nodes with 72 edges (out of 105 possible ones). The points a_n are taken randomly from the standard Gaussian distribution and the objective is to project $\mathbf{a}$ on the simplex and on the ℓ_1 ball with $\mathsf{s} = 1$. We chose $\rho = 1$ and initialized all vectors to zero, which appears to be generally a good choice.

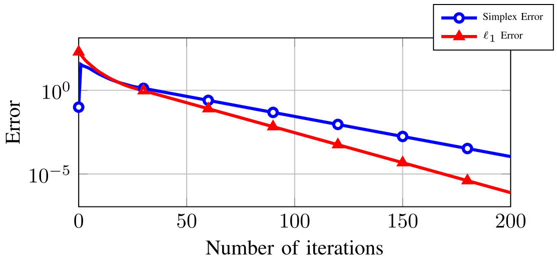


Fig. 3. Simplex projection error $\|\mathbf{p}^k - \mathcal{P}_\Delta(\mathbf{a})\|^2$ and ℓ_1 projection error $\|\mathbf{b}^k - \mathcal{P}_{\mathsf{B}}(\mathbf{a})\|^2$ with respect to the number of iterations.

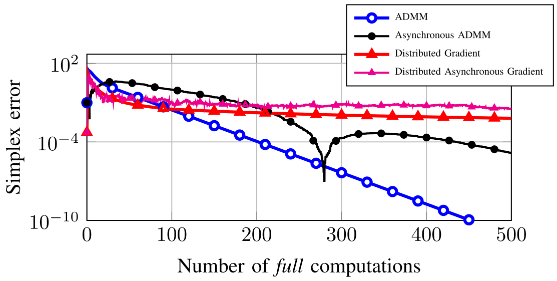


Fig. 4. Comparison between ADMM and Distributed Gradient.

In Fig. 2, we represent the agent threshold values (x_n^k) and projection values (p_n^k) with respect to the number of iterations for both problems. We notice that in these results, the agents quickly agree in terms of threshold values before shifting this value to the optimal one; this agreement/minimization trade-off is controlled by ρ (see [24, Fig. 2] for details).

In Fig. 3, we plot the projection errors ($\|\mathbf{p}^k - \mathcal{P}_\Delta(\mathbf{a})\|^2$ for the Simplex and $\|\mathbf{b}^k - \mathcal{P}_{\mathsf{B}}(\mathbf{a})\|^2$ for the ℓ_1-ball). They both converge linearly at comparable rates. Finally, in Fig. 4, we compare the ADMM approach of *Distributed Projection on the Simplex* with its asynchronous counterpart following [24]; we also add to plots the results of the classical distributed gradient and its asynchronous counterpart both with stepsizes $1/(1 + k/10)$ [32]. Out of fairness between asynchronous and synchronous methods, we plot the Simplex projection Error $\|\mathbf{p}^k - \mathcal{P}_\Delta(\mathbf{a})\|^2$ versus the number of *full* computations i.e. 1 per iteration for synchronous methods and $2/N$ per iteration for asynchronous ones. We notice that, as advocated, the ADMM-based approach is much more efficient than the gradient-based one on this problem; in addition, the asynchronous version enjoys a satisfying convergence rate.

VI. Conclusion

We proposed a framework to compute the projection onto the simplex or the ℓ_1 ball of a vector, the elements of which are stored and processed by the agents of a network in a distributed way. These methods are based on the careful construction of an optimization problem which writes as a sum of prox-easy functions and on distributed versions of the ADMM.

References

[1] M. H. DeGroot, "Reaching a consensus," *J. Amer. Statist. Assoc.*, vol. 69, no. 345, pp. 118–121, 1974.

[2] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. 31, no. 9, pp. 803–812, Sep. 1986.

[3] A. Demers *et al.*, "Epidemic algorithms for replicated database maintenance," in *Proc. 6th Annu. ACM Symp. Princ. Distrib. Comput.*, 1987, pp. 1–12.

[4] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.

[5] A. D. G. Dimakis, A. D. Sarwate, and M. J. Wainwright, "Geographic gossip: Efficient averaging for sensor networks," *IEEE Trans. Signal Process.*, vol. 56, no. 3, pp. 1205–1216, Mar. 2008.

[6] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc WSNs with noisy links—Part I: Distributed estimation of deterministic signals," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 350–364, Jan. 2008.

[7] J. Cortés, "Distributed algorithms for reaching consensus on general functions," *Automatica*, vol. 44, no. 3, pp. 726–737, 2008.

[8] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2748–2761, Jul. 2009.

[9] F. Iutzeler, P. Ciblat, and J. Jakubowicz, "Analysis of max-consensus algorithms in wireless channels," *IEEE Trans. Signal Process.*, vol. 60, no. 11, pp. 6103–6107, Nov. 2012.

[10] G. Giakkoupis and T. Sauerwald, "Rumor spreading and vertex expansion," in *Proc. 23rd Annu. ACM-SIAM Symp. Discrete Algorithms*, 2012, pp. 1623–1641.

[11] F. Iutzeler, P. Ciblat, and W. Hachem, "Analysis of sum-weight-like algorithms for averaging in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 61, no. 11, pp. 2802–2814, Jun. 2013.

[12] P. Bianchi and J. Jakubowicz, "Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization," *IEEE Trans. Autom. Control*, vol. 58, no. 2, pp. 391–405, Feb. 2013.

[13] S. Wu and M. G. Rabbat, "Broadcast gossip algorithms for consensus on strongly connected digraphs," *IEEE Trans. Signal Process.*, vol. 61, no. 16, pp. 3959–3971, Aug. 2013.

[14] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[15] E. Wei and A. Ozdaglar, "Distributed alternating direction method of multipliers," in *Proc. 51st IEEE Conf. Decis. Control*, 2012, pp. 5445–5450.

[16] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, "Asynchronous distributed optimization using a randomized alternating direction method of multipliers," in *Proc. 52nd IEEE Conf. Decis. Control*, 2013, pp. 3671–3676.

[17] E. Wei and A. Ozdaglar, "On the O(1/k) convergence of asynchronous distributed alternating direction method of multipliers," in *Proc. IEEE Global Conf. Signal Inf. Process.*, 2013, pp. 551–554.

[18] P. L. Lions and B. Mercier, "Splitting algorithms for the sum of two nonlinear operators," *SIAM J. Numer. Anal.*, vol. 16, no. 6, pp. 964–979, 1979.

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

[20] P. L. Combettes and J.-C. Pesquet, "Stochastic quasi-Fejér block-coordinate fixed point iterations with random sweeping," *SIAM J. Optim.*, vol. 25, no. 2, pp. 1221–1248, 2015.

[21] A. Chambolle and T. Pock, "A first-order primal–dual algorithm for convex problems with applications to imaging," *J. Math. Imag. Vision*, vol. 40, no. 1, pp. 120–145, 2011.

[22] L. Condat, "A primal–dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms," *J. Optim. Theory Appl.*, vol. 158, no. 2, pp. 460–479, 2013.

[23] P. Bianchi, W. Hachem, and F. Iutzeler, "A coordinate descent primal–dual algorithm and application to distributed asynchronous optimization," *IEEE Trans. Autom. Control*, vol. 61, no. 10, pp. 2947–2957, Oct. 2016.

[24] F. Iutzeler, "Distributed computation of quantiles via ADMM," *IEEE Signal Process. Lett.*, vol. 24, no. 5, pp. 619–623, May 2017.

[25] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, "Explicit convergence rate of a distributed alternating direction method of multipliers," *IEEE Trans. Autom. Control*, vol. 61, no. 4, pp. 892–904, Dec. 2013.

[26] L. Condat, "A convex approach to K-means clustering and image segmentation," in *Proc. 11th Int. Conf. Energy Minimization Methods Comput. Vision Pattern Recognit., (EMMCVPR)*, Lecture Notes in Computer Science, Venice, Italy, Oct. 2017, vol. 10746, pp. 220–234.

[27] A. Tiard, L. Condat, L. Drumetz, J. Chanussot, W. Yin, and X. Zhu, "Robust linear unmixing with enhanced sparsity," in *Proc. IEEE Int. Conf. Image Process., (ICIP)*, Beijing, China, Sept. 2017, pp. 3140–3144.

[28] L. Condat, "Fast projection onto the simplex and the ℓ_1 ball," *Math. Program. Series A*, vol. 158, no. 1, pp. 575–585, 2016.

[29] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the ℓ_1-ball for learning in high dimensions," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 272–279.

[30] M. Held, P. Wolfe, and H. P. Crowder, "Validation of subgradient optimization," *Math. Program.*, vol. 6, no. 1, pp. 62–88, 1974.

[31] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Cham, Switzerland: Springer, 2011.

[32] P. Bianchi, G. Fort, and W. Hachem, "Performance of a distributed stochastic approximation algorithm," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7405–7418, Nov. 2013.