2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

Distributed Optimization over Lossy Networks via Relaxed Peaceman-Rachford Splitting: a Robust ADMM Approach

N. Bastianello, M. Todescato, R. Carli, L. Schenato

Abstract— **In this work we address the problem of distributed optimization of the sum of convex cost functions in the context of multi-agent systems over lossy communication networks. Building upon operator theory, first, we derive an ADMM-like algorithm, referred to as relaxed ADMM (R-ADMM) via a generalized *Peaceman-Rachford Splitting* operator on the Lagrange dual formulation of the original optimization problem. This algorithm depends on two parameters, namely the averaging coefficient α and the augmented Lagrangian coefficient ρ and we show that by setting $\alpha = 1/2$ we recover the standard ADMM algorithm as a special case. Moreover, first, we reformulate our R-ADMM algorithm into an implementation that presents reduced complexity in terms of memory, communication and computational requirements. Second, we propose a further reformulation which let us provide the first ADMM-like algorithm with guaranteed convergence properties even in the presence of lossy communication. Finally, this work is complemented with a set of compelling numerical simulations of the proposed algorithms over random geometric graphs subject to i.i.d. random packet losses.**

Index Terms— **distributed optimization, ADMM, operator theory, splitting methods, Peaceman-Rachford operator**

I. Introduction

From classical control theory to more recent Machine Learning applications, many problems can be cast as optimization problems [1] and, in particular, as large-scale optimization problems given the advent of Internet-of-Things we are witnessing with its ever-increasing growth of large-scale cyber-physical systems. Hence, stemming from classical optimization theory, in order to break down the computational complexity, parallel and distributed optimization methods have been the focus of a wide branch of research [2]. Within this vast topic, typical applications, going under the name of *distributed consensus optimization*, foresee distributed computing nodes to communicate in order to achieve a desired common goal. More formally, the distributed nodes seek to

$$\min_x \sum_{i=1}^{N} f_i(x)$$

where, usually, each f_i is owned by one node only. Toward this application among very many different optimization algorithms explored in past as well as in current literature, e.g. subgradient methods [3], the well known Alternating Direction Method of Multipliers (ADMM), first introduced in [4] and [5], is recently receiving an ever-increasing interest because of its numerical efficiency and its natural structure which makes it well-suited for distributed and parallel computing. For an overview of possible applications and convergence results we refer the interested reader to [6], [7], [8], [9], [10], [11].

While ADMM can be proficiently applied to distributed setups, rigorous convergence results are usually provided under the assumptions of synchronous updates and lossless communications, which in practice are seldom verified.

On the one hand, an extensive body of work has been devoted to overcoming this limitation by adapting the ADMM to operate in an asynchronous fashion. For instance the work [12] proves convergence of the ADMM when the update of a random subset of coordinates is carried out at each instant. The papers [13], [14] analyze the convergence and convergence rate of a partially asynchronous ADMM – i.e., subject to a maximum allowed delay – for a master-slave architecture. Finally [15] defines a framework for asynchronous operations used to solve a broad class of optimization problems and showing how to derive an asynchronous ADMM formulation.

On the other hand, to the best of our knowledge, no work explicitly focuses on the robustness of ADMM to packet losses. The robustness to packet losses however has been studied for the Newton-Raphson consensus algorithm, see [16], [17].

In the following we resort to the body of literature on (nonexpansive) operator theory to carry out our analysis. Here, the underlying idea is to convert optimization problems into the problem of finding the fixed points of suitable nonexpansive operators [18]. Since, depending on the optimization problem, these fixed point problems might be unwieldy, the class of *splitting methods* has been developed, which exploit the problem's structure to break it in smaller and more manageable pieces. It is in the framework of splitting operators, and in particular the well recognized Peaceman-Rachford (PRS) [19] and Douglas-Rachford (DRS) [20], [21] splitting, that the ADMM comes into place. Indeed, the ADMM naturally arises as application of these operators to the Lagrange dual problem of the original optimization problem [22].

The remainder of the paper is organized as follows. Section II reviews the classical ADMM algorithm and its generalized version, introducing the necessary background on operator theory and splitting methods. Section III focuses on the analysis of distributed consensus optimization. Section IV analyzes the case with lossy communication. Section V collects some numerical simulations. Finally, Section VI draws some concluding remarks. Due to space constraints,

The authors are with the Department of Information Engineering, University of Padova, via Gradenigo 6/b 35131, Padova, Italy. nicola.bastianello.3@studenti.unipd.it, [todescat|carlirug|schenato]@dei.unipd.it

all the technical proofs can be found in the Appendix of [23].

II. Background on the ADMM

In this Section, we first review the popular ADMM algorithm [5], [6], then we introduce the more general *relaxed* ADMM (R-ADMM) algorithm and some notions of operator theory necessary to derive it. Finally we compare the two versions of ADMM.

A. The ADMM Algorithm

Consider the following optimization problem

$$\begin{aligned}&\min_{x\in\mathcal{X},y\in\mathcal{Y}}\{f(x)+g(y)\}\\ &\text{s.t. } Ax+By=c\end{aligned} \tag{1}$$

where $\mathcal{X}$ and $\mathcal{Y}$ are Hilbert spaces, $f:\mathcal{X}\to\mathbb{R}\cup\{+\infty\}$ and $g:\mathcal{Y}\to\mathbb{R}\cup\{+\infty\}$ are closed, proper and convex functions[1].
To solve problem (1) via the ADMM algorithm, we first define the *augmented Lagrangian* as

$$\begin{aligned}\mathcal{L}_\rho(x,y;w)=&\,f(x)+g(y)-w^\top(Ax+By-c)\\ &+\frac{\rho}{2}\|Ax+By-c\|^2\end{aligned} \tag{2}$$

where $\rho>0$ and w is the vector of Lagrange multipliers. The ADMM algorithm consists in keeping alternating the following update equations

$$y(k+1)=\arg\min_y \mathcal{L}_\rho(x(k),y;w(k)) \tag{3}$$
$$w(k+1)=w(k)-\rho(Ax(k)+By(k+1)-c) \tag{4}$$
$$x(k+1)=\arg\min_x \mathcal{L}_\rho(x,y(k+1);w(k+1)). \tag{5}$$

Notice that the above formulation is equivalent to the one proposed in [6] except for a change in the order of the updates which however does not affect the convergence properties of the algorithm. Moreover, the ADMM algorithm is provably shown to converge to the optimal solution of (1) for any $\rho>0$ assuming that $\mathcal{L}_0$ has a saddle point [6].

B. The Relaxed ADMM

The ADMM described above however is only a particular case of the so-called *relaxed ADMM*, which is derived from the application of the *relaxed Peaceman-Rachford splitting* (R-PRS), which we shall briefly introduce in the following, to the dual of problem (1).
First of all, though, we recall the following definition that will be useful.

Definition 1 (Proximal and reflective operators): Let $\mathcal{X}$ be a Hilbert space and $f:\mathcal{X}\to\mathbb{R}\cup\{+\infty\}$ be a closed, proper and convex function. We define the *proximal operator* of f with penalty $\rho>0$, $\operatorname{prox}_{\rho f}:\mathcal{X}\to\mathcal{X}$, as

$$\operatorname{prox}_{\rho f}(y)=\arg\min_{x\in\mathcal{X}}\left\{f(x)+\frac{1}{2\rho}\|x-y\|^2\right\}.$$

Moreover, we define the relative *reflective operator* as $\operatorname{refl}_{\rho f}=2\operatorname{prox}_{\rho f}-I$. □

Consider now the optimization problem $\min_{x\in\mathcal{X}}\{f(x)+g(x)\}$ with f, g closed, proper, convex and not necessarily smooth functions. We define the *Peaceman-Rachford operator* as

$$T_{PRS}=\operatorname{refl}_{\rho f}\circ\operatorname{refl}_{\rho g}$$

and hence the the R-PRS consists of the following iteration

$$z(k+1)=(1-\alpha)z(k)+\alpha T_{PRS}z(k) \tag{6}$$

where z is an auxiliary variable and the optimum to the problem can be recovered from the limit z^* of the iterate $z(k)$ as $x^*=\operatorname{prox}_{\rho g}(z^*)$.
An implementation of the R-PRS is given by the following equations

$$\psi(k)=\operatorname{prox}_{\rho g}(z(k)) \tag{7}$$
$$\xi(k)=\operatorname{prox}_{\rho f}(2\psi(k)-z(k)) \tag{8}$$
$$z(k+1)=z(k)+2\alpha(\xi(k)-\psi(k)) \tag{9}$$

where notice that the task of computing the proximal operators at the current iterate is split between equations (7) and (8), hence the name.
Since the PR operator is *nonexpansive*, i.e. it has Lipschitz constant equal to one, it is possible to prove that the R-PRS converges to a fixed point of T_{PRS} [18].

As mentioned above, the relaxed ADMM (R-ADMM) algorithm can be derived applying the R-PRS method to the Lagrange dual of problem (1), that is to

$$\min_{w\in\mathcal{W}}\{d_f(w)+d_g(w)\} \tag{10}$$

where

$$\begin{aligned}d_f(w)&=f^*(A^\top w)\\ d_g(w)&=g^*(B^\top w)-w^\top c,\end{aligned}$$

and f^*, g^* are the convex conjugates of f and g[2]. The derivation of problem (10) can be found in [24], [15].
Observe that, given the structure of problem (1) (i.e., proper closed and convex functions and linear constraints) there is no duality gap and, in turn, the optimal solutions of (1) and of (10) attain the same optimal value.
The motivation for dealing with the Lagrange dual problem relies on the fact that the minimization in (10) is performed over a single variable, thus allowing for the use of the R-PRS algorithm described in (7), (8) and (9).
Lemma 11 in [24] shows now that the update (7) and the update (8), applied to the dual problem, can be conveniently computed by, respectively,

$$\begin{aligned}&y(k)=\arg\min_y\left\{g(y)-z^\top(k)(By-c)+\frac{\rho}{2}\|By-c\|^2\right\}\\ &\psi(k)=z(k)-\rho(By(k)-c)\end{aligned} \tag{11}$$

[1]A function $f:\mathcal{X}\to\mathbb{R}\cup\{+\infty\}$ is said to be *closed* if $\forall a\in\mathbb{R}$ the set $\{x\in\operatorname{dom}(f)\mid f(x)\le a\}$ is closed. Moreover, f is said to be *proper* if it does not attain $-\infty$ [6].

[2]The *convex conjugate* of a function f is defined as $f^*(y)=\sup_{x\in\mathcal{X}}\{\langle y,x\rangle-f(x)\}$.

and

$$x(k) = \arg\min_{x} \left\{ f(x) - (2\psi(k) - z(k))^\top Ax + \frac{\rho}{2}\|Ax\|^2 \right\}$$
$$\xi(k) = 2\psi(k) - z(k) - \rho Ax(k) \tag{12}$$

The R-ADMM consists therefore in applying iteratively the set of five equations given by the two equations in (11), the two equations in (12) and equation (9).

A more compact formulation of the R-ADMM can be now be derived, which relates it to the ADMM of equations (3)–(5). Indeed it can be shown that the following updates are equivalent to the five equations of the R-ADMM (11), (12) and equation (9) [24].

$$\begin{aligned} y(k+1) = \arg\min_{y} \{ & \mathcal{L}_\rho(x(k), y; w(k)) \\ & + \rho(2\alpha - 1)\langle By, (Ax(k) + By(k) - c)\rangle \} \end{aligned} \tag{13}$$
$$\begin{aligned} w(k+1) = w(k) & - \rho(Ax(k) + By(k+1) - c) \\ & - \rho(2\alpha - 1)(Ax(k) + By(k) - c) \end{aligned} \tag{14}$$
$$x(k+1) = \arg\min_{x} \mathcal{L}_\rho(x, y(k+1); w(k+1)). \tag{15}$$

The ADMM algorithm in (3)–(5) can be recovered from this formulation of the R-ADMM by setting $\alpha = 1/2$, which cancels the additional terms weighted by $2\alpha - 1$.
It is of notice that the R-ADMM has two tunable parameters, ρ and α, against the only one of the ADMM, ρ, which is the cause of the greater reliability of the R-ADMM. Moreover, suitably choosing α can lead to larger convergence rates, see Section V.

III. Distributed Consensus Optimization

This Section introduces the distributed consensus convex optimization problem that we are interested in, and the solutions obtained by applying the R-ADMM algorithm.

A. Problem Formulation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph, with $\mathcal{V}$ the set of N vertices, labeled 1 through N, and $\mathcal{E}$ the set of undirected edges. For $i \in \mathcal{V}$, by $\mathcal{N}_i$ we denote the set of neighbors of node i in $\mathcal{G}$, namely,

$$\mathcal{N}_i = \{ j \in V \,:\, (i,j) \in \mathcal{E} \}.$$

We are interested in solving the following optimization problem

$$\min_{x} \sum_{i=1}^{N} f_i(x) \tag{16}$$

where $f_i : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ are closed, proper and convex functions and where f_i is known only to node i. In the following we denote by x^* the optimal solution of (16).
Observe that (16) can be equivalently formulated as

$$\begin{aligned} & \min_{x_i, \forall i} \sum_{i=1}^{N} f_i(x_i) \\ & \text{s.t. } x_i = x_j, \ \forall (i,j) \in \mathcal{E} \end{aligned} \tag{17}$$

By introducing for each edge $(i,j) \in \mathcal{E}$ the two *bridge variables* y_{ij} and y_{ji}, the constraints in (17) can be rewritten as

$$\begin{aligned} x_i &= y_{ij} \\ x_j &= y_{ji} \quad \forall (i,j) \in \mathcal{E}. \\ y_{ij} &= y_{ji} \end{aligned}$$

Defining $\mathbf{x} = [x_1^\top, \dots, x_N^\top]^\top$, $f(\mathbf{x}) = \sum_i f_i(x_i)$, and stacking all bridge variables in $\mathbf{y} \in \mathbb{R}^{n|\mathcal{E}|}$, we can reformulate the problem as

$$\begin{aligned} & \min_{\mathbf{x}} f(\mathbf{x}) \\ & \text{s.t. } \ A\mathbf{x} + \mathbf{y} = 0 \\ & \mathbf{y} = P\mathbf{y} \end{aligned}$$

for a suitable A matrix and with P being a permutation matrix that swaps y_{ij} with y_{ji}. Making use of the indicator function $\iota_{(I-P)}(\mathbf{y})$ which is equal to 0 if $(I-P)\mathbf{y} = 0$, and $+\infty$ otherwise, we can finally rewrite problem (17) as

$$\begin{aligned} & \min_{\mathbf{x}, \mathbf{y}} \{ f(\mathbf{x}) + \iota_{(I-P)}(\mathbf{y}) \} \\ & \text{s.t. } \ A\mathbf{x} + \mathbf{y} = 0. \end{aligned} \tag{18}$$

In next Section we apply the R-ADMM algorithm described in Section II-B to the above problem.

B. R-ADMM for Distributed Convex Optimization

In this section we propose an implementation of the R-ADMM for distributed convex optimization problems.
A direct application of equations (13)–(15) to problem (18) leads to a quite cumbersome algorithm, that has high storage requirements. Therefore we provide an alternative algorithm which is derived directly from the application of the set of five equations in (11), (12) and (9) to the dual of problem (18).
First of all we introduce the dual variables w_{ij} and w_{ji} which are associated to the constraints $x_i = y_{ij}$ and $x_j = y_{ji}$. Therefore, since the vector z has the same dimension of the vector w, this implies the presence of also the variables z_{ij} and z_{ji} for any $(i,j) \in \mathcal{E}$.
We have the following Proposition.

Proposition 1: The implementation of the R-ADMM algorithm described in the set of five equations given in (11), (12) and (9) applied to the dual of problem (18), reduces to alternating between the following two updates

$$\begin{aligned} x_i(k) = \arg\min_{x_i} \Bigg\{ & f_i(x_i) - \Bigg(\sum_{j \in \mathcal{N}_i} z_{ji}^\top(k) \Bigg) x_i \\ & + \frac{\rho}{2} |\mathcal{N}_i| \|x_i\|^2 \Bigg\}, \end{aligned} \tag{19}$$

for all $i \in V$, and

$$\begin{aligned} z_{ij}(k+1) &= (1-\alpha) z_{ij}(k) - \alpha z_{ji}(k) + 2\alpha\rho x_i(k) \\ z_{ji}(k+1) &= (1-\alpha) z_{ji}(k) - \alpha z_{ij}(k) + 2\alpha\rho x_j(k) \end{aligned} \tag{20}$$

for all $(i,j) \in \mathcal{E}$. □

Remark 1: Observe that the reformulation of (11), (12) and (9) as in Proposition 1 is possible for the particular structure of Problem (18) and, in particular, for the structure of the constraints $Ax + y = 0$. In general, given a set of constraints $Ax + By = c$ being A, B and c generic matrices and vector, such reformulation might not be possible. $\square$

The previous proposition naturally suggests a distributed implementation of the R-ADMM in which each node i stores in its local memory the variables x_i and $z_{ij}, j \in \mathcal{N}_i$. Then, at each iteration of the algorithm, each node i first collects the variables $z_{ji}, j \in \mathcal{N}_i$; second, updates x_i and z_{ij} according to (19) and the first of (20), respectively; finally, it sends z_{ij} to $j \in \mathcal{N}_i$.
Differently to the natural implementation just briefly described, we present a slightly different implementation building upon the observation that each node i, to update x_i as in (19) requires the variables z_{ji} rather than z_{ij} for $j \in \mathcal{N}_i$. Consequently, we assume node i stores in its memory and is in charge for the update of $z_{ji}, j \in \mathcal{N}_i$. The implementation is described in Algorithm 1.

```
Input: Set the termination condition $K > 0$. For each
       node $i$, initialize $x_i(0)$ and $z_{ji}(0)$, $j \in \mathcal{N}_i$.
$k \leftarrow 0$;
while $k < K$ each agent $i$ do
    compute $x_i(k)$ according to (19);
    for all $j \in \mathcal{N}_i$, compute the quantity $q_{i\to j}$ as
        $q_{i\to j} = -z_{ji}(k) + 2\rho x_i(k)$        (21)
    for all $j \in \mathcal{N}_i$, transmit $q_{i\to j}$ to node $j$;
    gather $q_{j\to i}$ from each neighbor $j$;
    update $z_{ji}$ as
        $z_{ji}(k+1) = (1-\alpha) z_{ji}(k) + \alpha q_{j\to i}$;        (22)
    $k \leftarrow k+1$;
end
```

Algorithm 1: Modified distributed R-ADMM.

Finally we state the convergence properties of Algorithm 1, which follow from that of the R-PRS.

Proposition 2: Consider Algorithm 1. Let (α, ρ) be such that $0 < \alpha < 1$ and $\rho > 0$. Then, for any initial conditions, the trajectories $k \to x_i(k)$, $i \in V$, generated by Algorithm 1, converge to the optimal solution of (16), i.e.,

$$\lim_{k\to\infty} x_i(k) = x^*, \qquad \forall i \in \mathcal{V},$$

for any $x_i(0)$ and $z_{ji}(0)$, $j \in \mathcal{N}_i$. $\square$

IV. Distributed R-ADMM over lossy networks

The distributed algorithm illustrated in the previous section works under the standing assumption that the communication channels are reliable, that is, no packet losses occur. The goal of this section is to relax this communication requirement and, in particular, to show that Algorithm 1 still converges, under a probabilistic assumption on communication failures which is next stated.

Assumption 1: During any iteration of Algorithm 1, the communication from node i to node j can be lost with some probability p. $\square$

In order to describe the communication failure more precisely, we introduce the family of independent binary random variables $L_{ij}(k)$, $k = 0, 1, 2, \ldots$, $i \in \mathcal{V}$, $j \in \mathcal{N}_i$, such that[3]

$$\mathbb{P}[L_{ij} = 1] = p, \qquad \mathbb{P}[L_{ij} = 0] = 1 - p.$$

We emphasize the fact that independence is assumed among all $L_{ij}(k)$ as i, j and k vary. If the packet transmitted, during the k-th iteration by node i to node j is lost, then $L_{ij}(k) = 1$, otherwise $L_{ij}(k) = 0$.
In this lossy scenario, Algorithm 1 is modified as shown in Algorithm 2.

```
Input: Set the termination condition $K > 0$. For each
       node $i$, initialize $x_i(0)$ and $z_{ji}(0)$, $j \in \mathcal{N}_i$.
$k \leftarrow 0$;
while $k < K$ each agent $i$ do
    compute $x_i(k)$ according to (19);
    for all $j \in \mathcal{N}_i$, compute the quantity $q_{i\to j}$ as
        $q_{i\to j} = -z_{ji}(k) + 2\rho x_i(k)$        (23)
    for all $j \in \mathcal{N}_i$, transmit $q_{i\to j}$ to node $j$;
    if for $j \in \mathcal{N}_i$, $q_{j\to i}$ is received then
        update $z_{ji}$ as
            $z_{ji}(k+1) = (1-\alpha) z_{ji}(k) + \alpha q_{j\to i}$;        (24)
    end
    $k \leftarrow k+1$;
end
```

Algorithm 2: Robust distributed R-ADMM.

In this case, at k-th iteration node i updates x_i as in (19). Then, for $j \in \mathcal{N}_i$, it computes $q_{i\to j}$ as in (23) and transmits it to node j. If node j receives $q_{i\to j}$, then it updates z_{ij} as $z_{ij}(k+1) = (1-\alpha) z_{ij}(k) + \alpha q_{i\to j}$, otherwise z_{ij} remains unchanged, i.e., $z_{ij}(k+1) = z_{ij}(k)$. This last step can be compactly describes as

$$\begin{aligned} z_{ij}(k+1) = L_{ij}(k) z_{ij}(k) + \\ + (1 - L_{ij}(k)) \left((1-\alpha) z_{ij}(k) + \alpha q_{i\to j} \right) \end{aligned}$$

We have the following Proposition.

Proposition 3: Consider Algorithm 2 working under the scenario described in Assumption 1. Let (α, ρ) be such that $0 < \alpha < 1$ and $\rho > 0$. Then, for any initial conditions, the trajectories $k \to x_i(k)$, $i \in \mathcal{V}$, generated by Algorithm 2, converge almost surely to the optimal solution of (16), i.e.,

$$\lim_{k\to\infty} x_i(k) = x^*, \qquad \forall i \in \mathcal{V},$$

with probability one, for all $i \in \mathcal{V}$, for any $x_i(0)$ and $z_{ji}(0)$, $j \in \mathcal{N}_i$. $\square$

[3]We highlight that the results of this section can be extended to the case where the loss probability is different for each edge.

Notice that the idea behind Proposition 3 is to show that the randomized ADMM described above conforms to the stochastic formulation of the PR splitting, introduced in [25], [12], whose convergence result therefore holds. The details can be found in the technical note [23].

Remark 2: We have restricted the analysis to the case of synchronous communication since we were mainly interested in investigating the algorithm performance in the presence of packet losses. The practically more appealing asynchronous scenario will be the focus of future research. □

Remark 3: Observe that Proposition 2, for the case of reliable communications, and Proposition 3, regarding the lossy scenario, share exactly the same region of convergence in the space of the parameters. This means that Algorithm 1 remains provably convergent if $0 < \alpha < 1$ and $\rho > 0$ in both cases. However, observe that the result is not *necessary and sufficient* and, in particular, the convergence might hold also for value of $\alpha \geq 1$. Indeed, in the simulation Section V we show that, for the case of quadratic functions $f_i,\ i \in \mathcal{V}$, the region of attraction in parameter space is larger. Moreover, despite what suggested by the intuition, the larger the packet loss probability p, the larger the region of convergence. However, this increased region of stability is counterbalanced by a slower convergence rate of the algorithm. □

V. Simulations

In this section we provide some experimental simulations to test the proposed R-ADMM Algorithm 2 to solve distributed consensus optimization problems (16). We are particularly interested in showing the algorithm performances in the presence of packet losses in the communication among neighboring nodes. To simplify the numerical analysis we restrict to the case of quadratic cost functions of the form

$$f_i(x_i) = a_i x_i^2 + b_i x_i + c_i$$

where, in general, the quantities $a_i, b_i, c_i \in \mathbb{R}$ are different for each node i. In this case the update of the primal variables becomes linear and, in particular, Eq. (19) reduces to

$$x_i(k) = \frac{\sum_{j\in\mathcal{N}_i} z_{ji}(k) - b_i}{2a_i + \rho|\mathcal{N}_i|}\,.$$

We consider the family of random geometric graphs with $N = 10$ and communication radius $r = 0.1$[p.u.] in which two nodes are connected if and only if their relative distance is less that r. We perform a set of 100 Monte Carlo runs for different values of packet losses probability p, step size α and penalty parameters ρ.
First of all, for different values of packet loss probability p and for fixed values of step size $\alpha = 1$ and penalty $\rho = 1$, Figure 1 shows the evolution of the relative error

$$\log \frac{\|x(k) - x^*\|}{\|x^*\|}$$

computed with respect to the unique minimizer x^* and averaged over 100 Monte Carlo runs. As expected, the higher the packet loss probability, the smaller the rate of convergence. Indeed, failures in the communication among neighboring nodes negatively affect the computations.

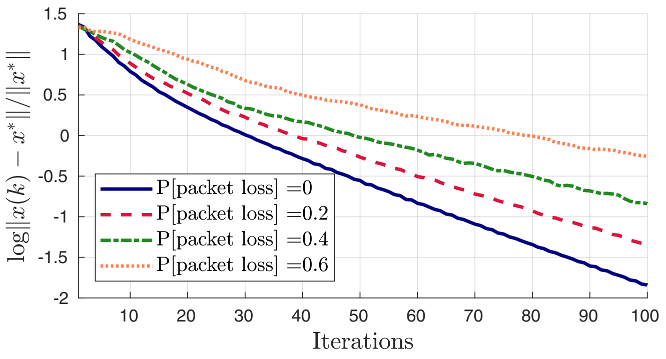

Fig. 1: Evolution, in log-scale, of the relative error of Alg. 2 computed w.r.t. the unique optimal solution x^* as function of different values of packet loss probability p for step size $\alpha = 1$ and penalty $\rho = 1$. Average over 100 Monte Carlo runs.

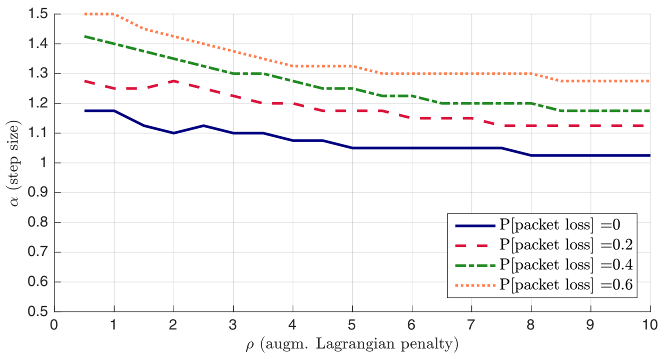

Fig. 2: Stability boundaries of Alg. 2 as function of the step size α and the penalty ρ for different values of loss probability p for the family of random geometric graphs. Average over 100 Monte Carlo runs.

Figure 2 plots the stability boundaries of the R-ADMM Algorithm 2 as function of step size α and penalty ρ for different packet loss probabilities p. More specifically, each curve in Figure 2 represents the numerical boundary below which the algorithm is found to be convergent and above which, conversely, the algorithm diverges. In this case the results turn out extremely interesting. Indeed, given α and ρ, for increasing packet loss probability p, the stability region enlarges. This means that the higher the loss probability is, the more robust the algorithm is. The numerical findings are perfectly in line with the result of Proposition 3, telling us that for $\alpha \in (0, 1)$ the algorithm converges for any value of ρ. However, it suggests the additional interesting fact that the theory misses to capture a larger area – in parameters space and depending on p – for which the algorithm still converges. This will certainly be a direction of future investigation.

Finally, Figure 3 reports the evolution of the error as a function of different values of the step-size α. Notice that to values of α that are larger than $1/2$ correspond faster convergences. Recalling that setting $\alpha = 1/2$ yields the standard ADMM, then it is clear that the use of the R-ADMM can speed up the convergence, which motivates its use against the use of the classic ADMM.

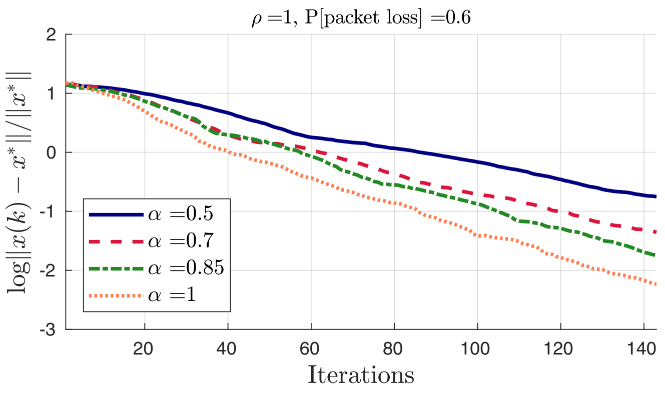

Fig. 3: Evolution, in log-scale, of the relative error of Alg. 2 computed w.r.t. the unique optimal solution x^* as function of different values of the step size α, with fixed packet loss probability $p = 0.6$ and penalty $\rho = 1$. Average over 100 Monte Carlo runs.

VI. Conclusions and Future Directions

In this paper we addressed the problem of distributed consensus optimization in the presence of synchronous but unreliable communications. Building upon results in operator theory on Hilbert spaces, we leveraged the relaxed Peaceman-Rachford Splitting operator to introduced what is referred to R-ADMM. We showed that the R-ADMM is a generalization of the well known ADMM. Then we introduced an algorithmic reformulation of the R-ADMM that has low computational, memory and communication requirements. Interestingly the algorithm, besides being extremely light from both the communication and memory point of views, turns out the be provably robust to random communication failures. Indeed, we rigorously proved how, in the lossy scenario, the region of convergence in parameters space remains unchanged compared to the case of reliable communication; yet, we numerically showed that the region of convergence is positively affected by a larger packet loss probability. The drawback lies in a slower convergence rate of the algorithm.
There remain many open questions paving the paths to future research directions such as analysis of the asynchronous case and generalization of the results to more general distributed optimization problems.

References

[1] K. Slavakis, G. B. Giannakis, and G. Mateos, "Modeling and optimization for big data analytics," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 18–31, 2014.

[2] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

[3] B. Johansson, M. Rabi, and M. Johansson, "A randomized incremental subgradient method for distributed optimization in networked systems," *SIAM Journal on Optimization*, vol. 20, no. 3, pp. 1157–1170, 2010.

[4] R. Glowinski and A. Marroco, "Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires," *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, vol. 9, no. R2, pp. 41–76, 1975. [Online]. Available: http://eudml.org/doc/193269

[5] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.

[6] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[7] M. Fukushima, "Application of the alternating direction method of multipliers to separable convex programming problems," *Computational Optimization and Applications*, vol. 1, no. 1, pp. 93–111, 1992.

[8] J. Eckstein and M. Fukushima, "Some reformulations and applications of the alternating direction method of multipliers," in *Large scale optimization*. Springer, 1994, pp. 115–134.

[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, no. 1, pp. 293–318, 1992.

[10] G. Chen and M. Teboulle, "A proximal-based decomposition method for convex minimization problems," *Mathematical Programming*, vol. 64, no. 1-3, pp. 81–101, 1994.

[11] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the alternating direction method of multipliers (admm): Quadratic problems," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 644–658, March 2015.

[12] P. Bianchi, W. Hachem, and F. Iutzeler, "A coordinate descent primal-dual algorithm and application to distributed asynchronous optimization," *IEEE Transactions on Automatic Control*, vol. 61, no. 10, pp. 2947–2957, 2016.

[13] R. Zhang and J. Kwok, "Asynchronous distributed admm for consensus optimization," in *International Conference on Machine Learning*, 2014, pp. 1701–1709.

[14] T.-H. Chang, M. Hong, W.-C. Liao, and X. Wang, "Asynchronous distributed admm for large-scale optimization—part i: algorithm and convergence analysis," *IEEE Transactions on Signal Processing*, vol. 64, no. 12, pp. 3118–3130, 2016.

[15] Z. Peng, Y. Xu, M. Yan, and W. Yin, "Arock: an algorithmic framework for asynchronous parallel coordinate updates," *SIAM Journal on Scientific Computing*, vol. 38, no. 5, pp. A2851–A2879, 2016.

[16] R. Carli, G. Notarstefano, L. Schenato, and D. Varagnolo, "Distributed quadratic programming under asynchronous and lossy communications via newton-raphson consensus," in *Control Conference (ECC), 2015 European*. IEEE, 2015, pp. 2514–2520.

[17] ——, "Analysis of newton-raphson consensus for multi-agent convex optimization under asynchronous and lossy communications," in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*. IEEE, 2015, pp. 418–424.

[18] H. H. Bauschke and P. L. Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, 2011, vol. 408.

[19] D. W. Peaceman and H. H. Rachford, Jr, "The numerical solution of parabolic and elliptic differential equations," *Journal of the Society for industrial and Applied Mathematics*, vol. 3, no. 1, pp. 28–41, 1955.

[20] J. Douglas and H. H. Rachford, "On the numerical solution of heat conduction problems in two and three space variables," *Transactions of the American mathematical Society*, vol. 82, no. 2, pp. 421–439, 1956.

[21] 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.

[22] J. Eckstein and W. Yao, "Augmented lagrangian and alternating direction methods for convex optimization: A tutorial and some illustrative computational results," *RUTCOR Research Reports*, vol. 32, 2012.

[23] N. Bastianello, M. Todescato, R. Carli, and L. Schenato, "Proof of robustness of the relaxed-PRS: a robust admm approach," University of Padova, Department of Information Engineering, Tech. Rep., 2017. [Online]. Available: http://automatica.dei.unipd.it/people/todescato/publications.html

[24] D. Davis and W. Yin, "Convergence rate analysis of several splitting schemes," in *Splitting Methods in Communication, Imaging, Science, and Engineering*. Springer, 2016, pp. 115–163.

[25] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, "Asynchronous distributed optimization using a randomized alternating direction method of multipliers," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*. IEEE, 2013, pp. 3671–3676.