Multi-agent structured optimization over message-passing architectures with bounded communication delays

Puya Latafat, Panagiotis Patrinos

***Abstract*— We consider the problem of solving structured convex optimization problems over a network of agents with communication delays. It is assumed that each agent performs its local updates using possibly outdated information from its neighbors under the assumption that the delay with respect to each neighbor is bounded but otherwise arbitrary. The private objective of each agent is represented by the sum of two possibly nonsmooth functions one of which is composed with a linear mapping. The global optimization problem consists of the aggregate of the local cost functions and a common Lipschitz-differentiable function. In the case when the coupling between agents is represented only through the common function, we employ the primal-dual algorithm proposed by Vũ and Condat. In the case when the linear maps introduce additional coupling between agents a new algorithm is developed. In both cases convergence is obtained under a strong convexity assumption. To the best of our knowledge, this is the first time that this form of delay is analyzed for a primal-dual algorithm in a message-passing local-memory model.**

I. Introduction

In this paper we consider a class of structured optimization problems that can be represented as follows:

$$\underset{x\in\mathbb{R}^n}{\mathbf{minimize}} \quad f(x)+\sum_{i=1}^{m}\big(g_i(x_i)+h_i(N_ix)\big), \tag{1}$$

where $x=(x_1,\dots,x_m)$, N_i is a linear mapping, h_i, g_i are proper closed convex (possibly) nonsmooth functions, and f is convex, continuously differentiable with Lipschitz continuous gradient. The goal is to solve (1) over a network of agents through local communications. Each agent is assumed to maintain its own private cost functions g_i and $h_i\circ N_i$, while f and (possibly) the linear mappings N_i represent the coupling between the agents. An important challenge in such a network is the assumption that the agents have access to the latest information required for their computations.

Most iterative algorithms for convex optimization can be written as

$$x^{k+1}=x^k-Tx^k, \tag{2}$$

where the mapping $\mathrm{Id}-T$ (Id is the identity operator) has some contractive property resulting in the convergence of the sequence to a zero of T. In distributed optimization the goal is to devise algorithms where a group of agents/processors distributively update certain coordinates of x while guaranteeing convergence to a zero of T.

There are two main computational models in distributed optimization (depicted in Fig. 1) with a range of hybrid models in between [1, Chap. 1]. These models are conceptually different and require different analysis. The model considered in this work is the local/private-memory model. Let us first describe the two models.

Shared-memory model: This model is characterized by the access of all agents/processors to a shared memory. A large body of literature exists for parallel coordinate descent algorithms for this problem. Typically, coordinate descent algorithms would require a memory lock to ensure consistent reading. Interesting recent works allow inconsistent reads [2], [3]. In this model, for the fixed point iteration (2), each processor reads the global memory and proceeds to choose a random coordinate $i\in\{1,\dots,m\}$ and to perform

$$x_i^{k+1}=x_i^k-T_i\hat{x}^k,$$

where $\hat{x}^k$ denotes the data loaded from the global memory to the local storage at the clock tick k, and T_i represents the operator that updates the ith coordinate. This form of updates are asynchronous in the sense that the processors update the global memory simultaneously resulting in possibly inconsistent local copy $\hat{x}^k$ due to other processors modifying the global memory during a read. The analysis of such algorithms would in general rely on either using the properties of the operator that updates the ith coordinate when possible (coordinate-wise Lipschitz continuity in the case of the gradient [2]), or the properties of the global operator (see [3] for nonexpansive operators). A crucial point in the convergence analysis of such methods is the fact that for a given processor, the index of the coordinate to be updated is selected at random, but no matter which coordinate is selected the same local data $\hat{x}^k$ is used for the update. Let $\hat{T}_ix\coloneqq(0,\dots,0,T_ix,0\dots,0)$. Then in a randomized scheme the operators $\hat{T}_i$ can be summed over i:

$$\sum_{i=1}^{m}\hat{T}_i\hat{x}^k=T\hat{x}^k,$$

allowing one to use the properties known for the global operator (see the proof of [3, Lem. 2]). As we discuss below, the difficulty in the local-memory model is precisely due to the fact that this summation no longer holds.

Local/private-memory model: In this model each agent/processor has its own private local memory. The agents can send and receive information to other agents as needed, and agent i can only update x_i. This model is also referred to as

Puya Latafat[1,2]; Email: puya.latafat@{kuleuven.be,imtlucca.it}

Panagiotis Patrinos[1]; Email: panos.patrinos@esat.kuleuven.be

This work was supported by: FWO PhD fellowship 1196818N; FWO projects: G086318N; G086518N; Fonds de la Recherche Scientifique – FNRS and the Fonds Wetenschappelijk Onderzoek – Vlaanderen under EOS Project no 30468160 (SeLMA)

[1]KU Leuven, Department of Electrical Engineering (ESAT-STADIUS), Kasteelpark Arenberg 10, 3001 Leuven-Heverlee, Belgium.

[2]IMT School for Advanced Studies Lucca, Piazza San Francesco 19, 55100 Lucca, Italy.

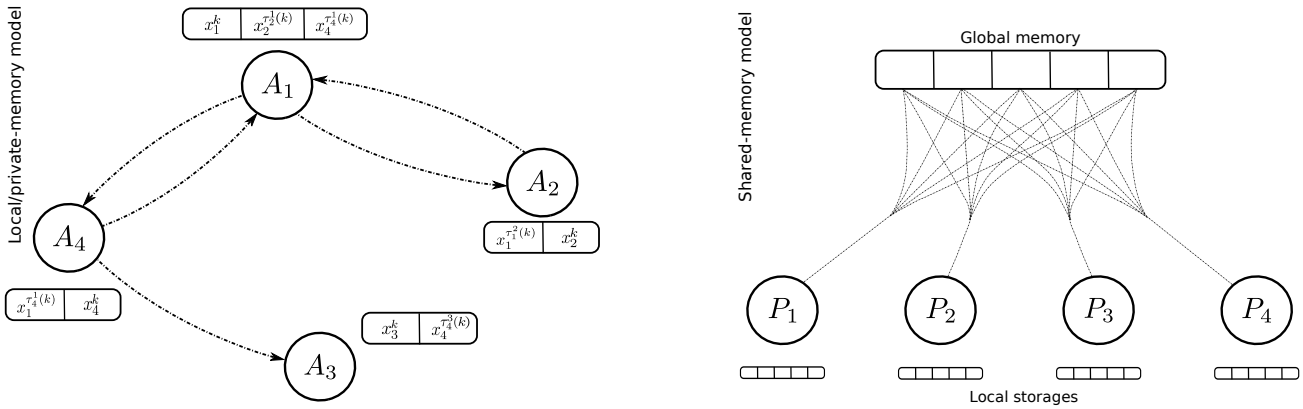

Fig. 1. The main two memory models; (left) agents cooperating to perform a task, (right) processors updating a global memory

message-passing model [1].

In the absence of delay between agents, randomized block-coordinate updates may be used to develop distributed asynchronous algorithms. Such schemes would typically involve random independent activation of agents to perform their local updates, and are in this sense also referred to as asynchronous [4]–[7]. In this work we are only concerned with the use of outdated information by the agents and do not pursue this form of asynchrony.

In accordance with the notation of the seminal work [1, Chap. 7] we define the following local (outdated) version of the generic vector $x^k = (x_1^k, \ldots, x_m^k)$ used by agent i:

$$x^k[i] := \left(x_1^{\tau_1^i(k)}, \ldots, x_m^{\tau_m^i(k)}\right), \tag{3}$$

where $\tau_j^i(k)$ is the latest time at which the value of x_j is transmitted to agent i by agent j. In our setting the delay is assumed to be bounded:

Assumption 1. *There exists an integer B such that for all $k \geq 0$ the following holds*

$$(\forall i, j) \quad 0 \leq k - \tau_j^i(k) \leq B, \quad \textit{and} \quad \tau_i^i(k) = k.$$

The fact that each agent knows its own local variable without delay is projected in the assumption $\tau_i^i(k) = k$. This is a natural assumption and is satisfied in practice. Notice that for ease of notation we defined the complete outdated vector while in practice each agent would only keep a local copy of the coordinates that are required for its computation, see Fig. 1. The direction of the arrows in Fig. 1 signify the nature of the coupling between two agents. For example, the arrow from A_4 to A_3 indicates that agent A_3 requires x_4 for its computation. Such a relation between agents is dependent on the formulation and the nature of coupling between agents. For instance, in the minimization (1), the coupling is represented through f and possibly N_i. As we shall see in §II the coupling through f may be one sided since agent i may require x_j for computing $\nabla_i f$ (the partial derivative of f with respect to x_i) without agent j requiring x_i.

In summary, each agent controls only one block of coordinates and updates according to

$$x_i^{k+1} = x_i^k - T_i x^k[i],$$

the result of which will be sent (possibly with delay) to the agents that require it in their computations. The difficulty in this model comes from the impossibility of summing $T_i x^k[i]$ over all i given that $x^k[i]$ is different for each i.

In addition to the above described delay, the *partially asynchronous* model considered in [1, Chap. 7] involves a second assumption: each agent must perform an update at least once during any time interval of a given length. Instead, we are not concerned with asynchrony but rather with the use of outdated information by the agents. We emphasize that developing *partially asynchronous* schemes for primal-dual algorithms or randomized schemes that comply with the delay model described in (3) remains a challenge.

In [1, Chap. 7.5] a partially asynchronous variant of the gradient method is studied. This analysis is further extended to the projected-gradient method in the convex case. In [8] a periodic linear convergence rate is established for the projected-gradient method. The recent work [9] extends this analysis to the proximal-gradient method. Notice that the aforementioned primal methods are not well equipped for problems with more complex structures as in (1).

In this work we study two primal-dual algorithms for solving (1) in the presence of bounded communication delays. Primal-dual proximal algorithms are a class of first-order methods that are easy to implement, are parallelizable, and yield the primal and dual solutions simultaneously. They are able to exploit the structure in (1) efficiently, resulting in *fully split* algorithms applicable to a wide range of applications. It is worth noting that while this paper focuses on two particular primal-dual algorithms, a similar analysis should be applicable to other primal-dual methods such as the ones developed in [6], [10]–[13].

A. Main Contributions

- To the best of our knowledge this is the first work that considers the general delay described in (3) for a primal-dual

algorithm. Unlike primal methods (gradient or proximal-gradient), this scheme can be applied to solve problems with complex structures as in (1) without the need to invert matrices or to solve inner loops.

- The analysis of [1], [8], [9] rely on the use of the cost as the Lyapunov function. In contrast, we show that under the bounded delay assumption and some strong convexity assumption, the generated sequence is quasi-Fejér monotone provided that the stepsizes are sufficiently small. Moreover, linear convergence is established with an explicit convergence factor.
- Two primal-dual algorithms are presented: (i) when the coupling between agents is enforced only through f, the algorithm of [14], [15] is considered, (ii) when the coupling is enforced through f and the linear mapping a modified algorithm is developed which appears to be new. In the second case due to the presence of additional coupling smaller stepsizes must be used to ensure convergence.

B. Motivating Example

Consider the problem of formation control [16], where each agent (vehicle) has its own private dynamics and cost function and the goal is to achieve a specific formation while communicating only with a selected number of agents. Let $w_i = (\xi_i, v_i)$ where ξ_i and v_i denote the local state and input sequences. The location of agent i is given by $y_i = C\xi_i$ and the set of its neighbors is denoted by $\mathcal{A}_i$. The linear dynamics of each agent over a control horizon is represented by the constraints $E_i w_i = b_i$. In order to enforce a formation between agents i and j the quadratic cost function $\|C(\xi_i - \xi_j) - d_{ij}\|^2$ is used where d_{ij} is the target relative distance between them (refer to [16] for details). Hence, the formation control problem is formulated as the following constrained minimization:

$$\begin{aligned}&\textbf{minimize} \ \tfrac{1}{2}\sum_{i=1}^{m}\sum_{j\in\mathcal{A}_i}\|C(\xi_i-\xi_j)-d_{ij}\|^2+\tfrac{1}{2}\sum_{i=1}^{m}w_i^\top Q_i w_i\\ &\textbf{subject to} \ E_i w_i = b_i,\ \ i=1,\dots,m\end{aligned}$$

This problem can be easily cast in the form of (1) by setting f equal to the first term, g_i equal to the quadratic local cost, h_i the indicator of the point b_i and the linear mapping $N_i = E_i$. Therefore, the objective is to enforce a formation between agents by solving this optimization problem in presence of communication delays by allowing the agents to use outdated information. Notice that in this case the coupling between agents is enforced only through f. This special case of (1) is studied in §III.

C. Notation and Preliminaries

Throughout, $\mathbb{R}^n$ is the n-dimensional Euclidean space with inner product $\langle\cdot,\cdot\rangle$ and induced norm $\|\cdot\|$. For a positive definite matrix P we define the scalar product $\langle x,y\rangle_P = \langle x,Py\rangle$ and the induced norm $\|x\|_P = \sqrt{\langle x,x\rangle_P}$.

For a set C, we denote its relative interior by $\mathbf{ri}C$. Let $q:\mathbb{R}^n\to\overline{\mathbb{R}} := \mathbb{R}\cup\{+\infty\}$ be a proper closed convex function. Its domain is denoted by $\mathbf{dom}\,q$. Its subdifferential is the set-valued operator $\partial q:\mathbb{R}^n\rightrightarrows\mathbb{R}^n$

$$\partial q(x) = \{y\in\mathbb{R}^n \mid \forall z\in\mathbb{R}^n, \langle z-x,y\rangle + f(x)\le f(z)\}.$$

For a positive scalar ρ the *proximal map* associated with q is the single-valued mapping defined by

$$\mathbf{prox}_{\rho q}(x) := \operatorname*{\mathbf{argmin}}_{z\in\mathbb{R}^n}\{q(z)+\tfrac{1}{2\rho}\|x-z\|^2\}.$$

The *Fenchel conjugate* of q, denoted by q^*, is defined as $q^*(v) := \mathbf{sup}_{x\in\mathbb{R}^n}\{\langle v,x\rangle - q(x)\}$. The function q is said to be μ-convex with $\mu\ge 0$ if $q(x) - \frac{\mu}{2}\|x\|^2$ is convex.

A sequence $(w^k)_{k\in\mathbb{N}}$ is said to be *quasi-Fejér* monotone relative to the set $\mathcal{U}$ if for all $v\in\mathcal{U}$ and all $k\in\mathbb{N}$

$$\|w^{k+1}-v\|^2 \le \|w^k - v\|^2 + \varepsilon^k,$$

where $(\varepsilon^k)_{k\in\mathbb{N}}$ is a summable nonnegative sequence [17]. The positive part of $x\in\mathbb{R}$ is denoted by $[x]_+ := \mathbf{max}\{x,0\}$.

II. Problem Setup

Throughout this paper the primal and dual vectors, denoted x and u, are assumed to be composed of m blocks as follows

$$x = (x_1,\dots,x_m)\in\mathbb{R}^n,\ u = (u_1,\dots,u_m)\in\mathbb{R}^r,$$

where $x_i\in\mathbb{R}^{n_i}$ and $u_i\in\mathbb{R}^{r_i}$. Consider a linear mapping $L:\mathbb{R}^n\to\mathbb{R}^r$ that is partitioned as follows:

$$L = \begin{pmatrix} L_{11} & \cdots & L_{1m}\\ \vdots & \ddots & \vdots\\ L_{m1} & \cdots & L_{mm}\end{pmatrix}, \tag{4}$$

where $L_{ij}:\mathbb{R}^{n_i}\to\mathbb{R}^{r_j}$. Furthermore, the ith (block) row of L is denoted by $L_{i\bullet}:\mathbb{R}^n\to\mathbb{R}^{r_i}$ and the ith (block) column by $L_{\bullet i}:\mathbb{R}^{n_i}\to\mathbb{R}^r$, *i.e.*,

$$L = \begin{pmatrix} L_{1\bullet}\\ \vdots\\ L_{m\bullet}\end{pmatrix} = \begin{pmatrix} L_{\bullet 1} & \cdots & L_{\bullet m}\end{pmatrix}.$$

The following holds

$$\langle Lx,u\rangle = \sum_{i=1}^{m}\langle L_{i\bullet}x,u_i\rangle = \sum_{i=1}^{m}\langle x_i, L_{\bullet i}^\top u\rangle. \tag{5}$$

Consider the structured optimization problem (1) where the linear mapping N_i has been replaced by $L_{i\bullet}$ defined above in order to clarify the structure of the mapping:

$$\operatorname*{\mathbf{minimize}}_{x\in\mathbb{R}^n} \ \ f(x)+\sum_{i=1}^{m}\big(g_i(x_i)+h_i(L_{i\bullet}x)\big). \tag{6}$$

The cost functions g_i and $h_i\circ L_{i\bullet}$ are private functions belonging to agent i. The coupling between agents is through the smooth term f and the linear term $L_{i\bullet}x$. An agent i is assumed to have access to the information required for its computation, be it outdated, *cf.* Algorithms 1 and 2.

Let the following assumptions hold

Assumption 2.

(i) For $i=1,\dots,m$, $g_i:\mathbb{R}^{n_i}\to\overline{\mathbb{R}}$, $h_i:\mathbb{R}^{r_i}\to\overline{\mathbb{R}}$ *are proper closed convex functions, and* $L_{i\bullet}:\mathbb{R}^n\to\mathbb{R}^{r_i}$ *is a linear mapping.*

(ii) $f:\mathbb{R}^n\to\mathbb{R}$ *is convex, continuously differentiable, and* ∇f *is* β*-Lipschitz continuous for some nonnegative* β*:*

$$\|\nabla f(x) - \nabla f(x')\| \le \beta\|x-x'\|,\ \ \forall x,x'\in\mathbb{R}^n.$$

(iii) For every $i=1,\dots,m$ there exists a nonnegative constant $\bar\beta_i$ such that for all $x,x'\in\mathbb{R}^n$ satisfying $x_i=x'_i$:

$$\|\nabla_i f(x)-\nabla_i f(x')\|\le\bar\beta_i\|x-x'\|. \tag{7}$$

(iv) The set of solutions to (6) is nonempty.

(v) (Constraint qualification) There exists $x_i\in\mathbf{ridom}\,g_i$, for $i=1,\dots,m$ such that $L_j x\in\mathbf{ridom}\,h_j$, for $j=1,\dots,m$.

Assumption 2*(iii)* quantifies the strength of the coupling (through f) between agents [1, Sec. 7.5]. In particular, if f is separable, *i.e.*, $f(x)=\sum_{i=1}^m f_i(x_i)$, then there is no coupling and $\bar\beta_i=0$.

Problem (6) can be compactly represented as

$$\underset{x\in\mathbb{R}^n}{\mathbf{minimize}}\quad f(x)+g(x)+h(Lx),$$

where $g(x)=\sum_{i=1}^m g_i(x_i)$, $h(u)=\sum_{i=1}^m h_i(u_i)$, and L is as in (4). The dual problem is given by

$$\underset{u\in\mathbb{R}^r}{\mathbf{minimize}}\ (g+f)^*(-L^\top u)+h^*(u).$$

Under the constraint qualification of Assumption 2*(v)*, the set of solutions to the dual problem is nonempty and the duality gap is zero [18, Cor. 31.2.1]. Furthermore, $x^\star$ is a primal solution and $u^\star$ is a dual solution if and only if the pair $(x^\star,u^\star)$ satisfies

$$\begin{cases}0\in\partial g(x^\star)+\nabla f(x^\star)+L^\top u^\star,\\ 0\in\partial h^*(u^\star)-Lx^\star.\end{cases} \tag{8}$$

Such a point is called a primal-dual solution and the set of all primal-dual solutions is denoted by $\mathcal{S}$.

Let us define a few parameters used throughout the paper. For each agent $i\in\{1,\dots,m\}$ define the positive stepsizes γ_i, σ_i that are associated with the primal and the dual variables, respectively. Moreover set

$$\begin{aligned}\bar\beta&:=(\bar\beta_1,\dots,\bar\beta_m),\\ \Gamma&:=\mathbf{blkdiag}(\gamma_1 I_{n_1},\dots,\gamma_m I_{n_m}),\\ \Sigma&:=\mathbf{blkdiag}(\sigma_1 I_{r_1},\dots,\sigma_m I_{r_m}).\end{aligned}$$

Applying the algorithm of Vũ and Condat [14], [15] to (6), with stepsize matrices Σ and Γ as defined above, results in the following updates for agent i at iteration k:

$$x_i^{k+1}=\mathbf{prox}_{\gamma_i g_i}\big(x_i^k-\gamma_i L_{\bullet i}^\top u^k-\gamma_i\nabla_i f(x^k)\big) \tag{9a}$$

$$u_i^{k+1}=\mathbf{prox}_{\sigma_i h_i^*}\big(u_i^k+\sigma_i L_{\bullet i}(2x^{k+1}-x^k)\big). \tag{9b}$$

Notice that each agent requires the latest variables x^k, x^{k+1} and u^k in the above updates, which may not be available due to communication delays. In the next section we consider the case when L is block-diagonal. The case of general L is studied in Section IV where a modified primal-dual algorithm is proposed in place of (9) to allow for a larger stepsize in this case.

III. The Case of Block-Diagonal Linear Mapping

Throughout this section we assume that the linear mapping L has a block-diagonal structure. Therefore, the coupling between agents is enacted only through the smooth function f. The example of formation control in Section I-B is of this structure.

Under this assumption problem (6) becomes

$$\underset{x\in\mathbb{R}^n}{\mathbf{minimize}}\quad f(x)+\sum_{i=1}^m\big(g_i(x_i)+h_i(L_{ii}x_i)\big),$$

where L_{ii} is the ith diagonal block of L, see (4). Given this diagonal structure, in the updates (9), agent i must receive those x_j's that are required for the computation of $\nabla_i f$ and all other operations are local. Let us define the set of agents that are required to send their variable to i as follows:

$$\mathcal{N}_i^{\text{in}}:=\{j\mid\nabla_i f\text{ depends on }x_j\},$$

and the set of j's that agent i must send x_i to as $\mathcal{N}_i^{\text{out}}:=\{j\mid i\in\mathcal{N}_j^{\text{in}}\}$.

Algorithm 1 summarizes the proposed scheme for this problem. At every iteration each agent i performs the updates described in (9) using the last information it has received from agents $j\in\mathcal{N}_i^{\text{in}}$. It then transmits the updated x_i^{k+1} to the agents that require it (possibly with delay). Note that $x^k[i]$ was defined as the outdated version of the full vector x^k for simplicity of notation, and in practical implementation it would only involve the coordinates that are required for the computation of $\nabla_i f$.

Algorithm 1 Vũ-Condat algorithm with bounded delays

Initialize: $x_i^0\in\mathbb{R}^{n_i}$, $u_i^0\in\mathbb{R}^{r_i}$ for each $i\in\{1,\dots,m\}$.

for $k=0,1,\dots$ **do**

for each agent $i=1,\dots,m$ **do**

– perform the local updates using the last received information, *i.e.*, using the locally stored vector $x^k[i]$ as defined in (3):

$x_i^{k+1}=\mathbf{prox}_{\gamma_i g_i}\big(x_i^k-\gamma_i L_{ii}^\top u_i^k-\gamma_i\nabla_i f(x^k[i])\big)$

$u_i^{k+1}=\mathbf{prox}_{\sigma_i h_i^*}\big(u_i^k+\sigma_i L_{ii}(2x_i^{k+1}-x_i^k)\big)$

– send x_i^{k+1} to all $j\in\mathcal{N}_i^{\text{out}}$ (possibly with different delays)

As shown in Theorem 1, for small enough stepsizes the generated sequence converges to a primal-dual solution under the bounded delay assumption, and provided that functions g_i are strongly convex. Such needed requirements are summarized below:

Assumption 3. *For $i=1,\dots,m$*

(i) (Strong convexity) g_i is μ_g^i-convex for some $\mu_g^i>0$.

(ii) (Convergence condition) The stepsizes $\sigma_i,\gamma_i>0$ satisfy the following assumption:

$$\gamma_i<\frac{1}{\sigma_i\|L_{ii}\|^2+\beta+\frac{B^2}{2}\|\bar\beta\|^2_{M_g^{-1}}}, \tag{10}$$

where

$$M_g=\mathbf{blkdiag}\big(\mu_g^1 I_{n_1},\dots,\mu_g^m I_{n_m}\big). \tag{11}$$

Notice that according to Assumption 3*(ii)* we require a one time global communication of $\|\bar\beta\|_{M_g^{-1}}$ and β when initiating the algorithm.

Before proceeding with the convergence results, let us define the following

$$P:=\begin{pmatrix}\Gamma^{-1}&-L^\top\\ -L&\Sigma^{-1}\end{pmatrix}. \tag{12}$$

Noting that Σ,Γ are positive definite, and using Schur complement we have that P is positive definite if and only if $\Gamma^{-1}-L^\top\Sigma L$ is positive definite, a condition that holds if (10) is satisfied (since L has a block-diagonal structure).

Our analysis in Theorem 1 relies on showing that the generated sequence is quasi-Fejér monotone relative to the set of primal-dual solutions in the space equipped with the inner product $\langle\cdot,\cdot\rangle_P$. Notice that without communication delays ($B\equiv 0$), this analysis leads to the usual Fejér monotonicity of the sequence. The use of outdated information introduces additional error terms that are shown to be tolerated by the algorithm if the stepsizes are small enough and the functions g_i are strongly convex.

The proof of Theorem 1 can be found in [19].

Theorem 1. *Consider Algorithm 1 and let Assumptions 1 to 3 hold. Then the sequence $(z^k)_{k\in\mathbb{N}}$ is quasi-Fejér monotone relative to $\mathcal{S}$ in the space equipped with the inner product $\langle\cdot,\cdot\rangle_P$. Furthermore, $(z^k)_{k\in\mathbb{N}}$ converges to some $z^\star\in\mathcal{S}$.*

IV. The Case of General Linear Mapping

In this section we consider the general optimization problem (6) where additional coupling is present through the linear maps, *i.e.*, L is not block-diagonal. We consider a modified primal-dual algorithm that resembles (9) with the difference that in the dual update the linear map $L_{i\bullet}$ operates on $x^k[i]$ in place of $2x^{k+1}[i]-x^k[i]$. This modification results in the possibility of using larger stepsizes since the terms $2x^{k+1}[i]-x^k[i]$ would introduce additional sources of error.

Let us define the following two sets:

$$\mathcal{M}_i^{\mathrm{p}} := \{j \mid L_{ji}\neq 0\},\ \ \mathcal{M}_i^{\mathrm{d}} := \{j \mid L_{ij}\neq 0\},$$

where 0 denotes a zero matrix of appropriate dimensions. In Algorithm 2, due to the additional coupling through the linear maps, the primal vector of agent i must be transmitted to all $j\in\mathcal{M}_i^{\mathrm{p}}\cup\mathcal{N}_i^{\mathrm{out}}$ while the dual vector is to be transmitted to all $j\in\mathcal{M}_i^{\mathrm{d}}$. Notice that the outdated primal and dual vectors $x^k[i]$ and $u^k[i]$, need not have the same delay pattern and are arbitrary as long as Assumption 1 is satisfied, *i.e.*, agent i may use the primal vector $x_j^{k_1}$ and the dual vector $u_j^{k_2}$ transmitted by j at times k_1 and k_2.

Algorithm 2 A primal-dual algorithm with bounded delays

Initialize: $x_i^0\in\mathbb{R}^{n_i}$, $u_i^0\in\mathbb{R}^{r_i}$ for each $i\in\{1,\dots,m\}$.
for $k=0,1,\dots$ **do**
  for each agent $i=1,\dots,m$ **do**
    – perform the local updates using the last received information, *i.e.*, using the locally stored vectors $x^k[i]$ and $u^k[i]$ as defined in (3):
    $x_i^{k+1}=\mathbf{prox}_{\gamma_i g_i}\big(x_i^k-\gamma_i L_{\bullet i}^\top u^k[i]-\gamma_i\nabla_i f(x^k[i])\big)$
    $u_i^{k+1}=\mathbf{prox}_{\sigma_i h_i^\star}\big(u_i^k+\sigma_i L_{i\bullet}x^k[i]\big)$
    – send x_i^{k+1} to all $j\in\mathcal{N}_i^{\mathrm{out}}\cup\mathcal{M}_i^{\mathrm{p}}$, and u_i^{k+1} to all $j\in\mathcal{M}_i^{\mathrm{d}}$ (possibly with different delays)

In Theorem 2 convergence is established for Algorithm 2 when the stepsizes are small enough, under the assumption that the functions g_i are strongly convex and h_i are continuously differentiable with Lipschitz continuous gradient. We summarize these requirements below:

Assumption 4. *For all $i=1,\dots,m$:*

(i) (Strong convexity) g_i is μ_g^i-convex for some $\mu_g^i>0$.

(ii) (Lipschitz continuity) h_i is continuously differentiable, and ∇h_i is $\frac{1}{\mu_h^i}$-Lipschitz continuous for some $\mu_h^i>0$. Equivalently, h_i^* is μ_h^i-convex.

(iii) (Convergence condition) The stepsizes $\sigma_i,\gamma_i>0$ satisfy the following inequalities

$$\sigma_i<\frac{1}{C_s(B+1)^2},\ \ \gamma_i<\frac{1}{\beta+\frac{1}{2}R_s(B+1)^2+B^2\|\bar\beta\|^2_{M_g^{-1}}},$$

where

$$R_s:=\sum_{i=1}^m\tfrac{1}{\mu_h^i}\|L_{i\bullet}\|^2,\ \ C_s:=\sum_{i=1}^m\tfrac{1}{\mu_g^i}\|L_{\bullet i}^\top\|^2. \tag{13}$$

Notice that by Assumption 4*(iii)* we require a one time global communication of R_s, C_s, β and $\|\bar\beta\|_{M_g^{-1}}$.

Let us define the following positive definite matrix that is used in the convergence analysis

$$D:=\mathbf{blkdiag}(\Gamma^{-1},\Sigma^{-1}). \tag{14}$$

We proceed with the convergence results for Algorithm 2. The proofs of Theorems 2 and 3 can be found in [19].

Theorem 2. *Consider Algorithm 2 and let Assumptions 1, 2 and 4 hold. Then the sequence $(z^k)_{k\in\mathbb{N}}$ is quasi-Fejér monotone relative to $\mathcal{S}$ in the space equipped with $\langle\cdot,\cdot\rangle_D$. Furthermore, $(z^k)_{k\in\mathbb{N}}$ converges to some $z^\star\in\mathcal{S}$.*

Next theorem provides a sufficient condition for the stepsizes under which linear convergence is attained.

Theorem 3 (Linear convergence)**.** *Consider Algorithm 2 and let Assumption 1, 2, 4(i) and 4(ii) hold. Let c be a positive scalar and set $\gamma_i=\frac{c}{\mu_g^i},\sigma_i=\frac{c}{\mu_h^i}$ for $i=1,\dots,m$. Let $\mu_g^{\min}=\mathbf{min}\{\mu_g^1,\dots,\mu_g^m\}$, $\mu_h^{\min}=\mathbf{min}\{\mu_h^1,\dots,\mu_h^m\}$. Suppose that the following holds:*

$$c\le(1+c_2)^{\frac{1}{B+1}}-1,$$

where

$$c_2=\mathbf{min}\left\{\frac{\mu_g^{\min}}{2B\|\bar\beta\|^2_{M_g^{-1}}+R_s(B+1)+\beta},\frac{\mu_h^{\min}}{2C_s(B+1)}\right\}.$$

Then the following linear convergence rate holds

$$\|z^k-z^\star\|^2\le\left(\tfrac{1}{1+c}\right)^k\|z^0-z^\star\|^2.$$

V. Conclusion & Future Works

In this paper we considered the application of primal-dual algorithms for solving structured optimization problems in a message-passing network model. It is shown that the communication delay is tolerated by the considered algorithms provided that the stepsizes are small enough, and that some strong convexity assumption holds. Future work consists of extending the convergence analysis to the partially asynchronous framework. Another research direction is to devise randomized schemes where in addition to the use of outdated

information, the agents would wake up at random independently from one another.

REFERENCES

[1] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Prentice-Hall, 1989, vol. 23.

[2] J. Liu and S. J. Wright, "Asynchronous stochastic coordinate descent: Parallelism and convergence properties," *SIAM Journal on Optimization*, vol. 25, no. 1, pp. 351–376, 2015.

[3] 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.

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

[5] 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, Oct 2016.

[6] P. Latafat, N. M. Freris, and P. Patrinos, "A new randomized block-coordinate primal-dual proximal algorithm for distributed optimization," *arXiv preprint arXiv:1706.02882*, 2017.

[7] J.-C. Pesquet and A. Repetti, "A class of randomized primal-dual algorithms for distributed optimization," *Journal of Nonlinear and Convex Analysis*, vol. 16, no. 12, pp. 2453–2490, 2015.

[8] P. Tseng, "On the rate of convergence of a partially asynchronous gradient projection algorithm," *SIAM Journal on Optimization*, vol. 1, no. 4, pp. 603–619, 1991.

[9] Y. Zhou, Y. Liang, Y. Yu, W. Dai, and E. P. Xing, "Distributed proximal gradient algorithm for partially asynchronous computer clusters," *Journal of Machine Learning Research*, vol. 19, no. 19, pp. 1–32, 2018.

[10] P. L. Combettes and J.-C. Pesquet, "Primal-dual splitting algorithm for solving inclusions with mixtures of composite, Lipschitzian, and parallel-sum type monotone operators," *Set-Valued and Variational Analysis*, vol. 20, no. 2, pp. 307–330, 2012.

[11] L. M. Briceño-Arias and P. L. Combettes, "A monotone + skew splitting model for composite monotone inclusions in duality," *SIAM Journal on Optimization*, vol. 21, no. 4, pp. 1230–1250, 2011.

[12] Y. Drori, S. Sabach, and M. Teboulle, "A simple algorithm for a class of nonsmooth convex-concave saddle-point problems," *Operations Research Letters*, vol. 43, no. 2, pp. 209–214, 2015.

[13] P. Latafat and P. Patrinos, "Asymmetric forward–backward–adjoint splitting for solving monotone inclusions involving three operators," *Computational Optimization and Applications*, pp. 1–37, 2017.

[14] L. Condat, "A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms," *Journal of Optimization Theory and Applications*, vol. 158, no. 2, pp. 460–479, 2013.

[15] B. C. Vũ, "A splitting algorithm for dual monotone inclusions involving cocoercive operators," *Advances in Computational Mathematics*, vol. 38, no. 3, pp. 667–681, 2013.

[16] R. L. Raffard, C. J. Tomlin, and S. P. Boyd, "Distributed optimization for cooperative agents: application to formation flight," in *43rd IEEE Conference on Decision and Control*, vol. 3, 2004, pp. 2453–2459.

[17] P. L. Combettes, "Quasi-Fejérian analysis of some optimization algorithms," *Studies in Computational Mathematics*, vol. 8, pp. 115–152, 2001.

[18] R. Rockafellar, *Convex analysis*. Princeton University Press, 1997.

[19] P. Latafat and P. Patrinos, "Multi-agent structured optimization over message-passing architectures with bounded communication delays," *arXiv preprint arXiv:1809.07199*, 2018.