Distributed ADMM With Linear Updates Over Directed Networks

Kiran Rokade and Rachel Kalpana Kalaimani

Abstract—Distributed optimization over a network of agents is ubiquitous in applications such as power system, robotics and statistical learning. In many settings, the communication network is directed, i.e., the communication links between agents are unidirectional. While several variations of gradient-descent-based primal methods have been proposed for distributed optimization over directed networks, an extension of dual-ascent methods to directed networks remains a less-explored area. In this paper, we propose a distributed version of the Alternating Direction Method of Multipliers (ADMM) with linear updates for directed networks using balancing weights, called BW-DADMM (Balancing Weights Directed ADMM). We show that if the objective function of the minimization problem is smooth and strongly convex, then BW-DADMM achieves a geometric rate of convergence to the optimal point. Our algorithm exploits the robustness inherent to ADMM by not enforcing accurate consensus, thereby significantly improving the convergence rate. We illustrate this by numerical examples, where we compare the performance of BW-DADMM with that of state-of-the-art ADMM methods over directed graphs.

Index Terms—Alternating Direction Method of Multipliers (ADMM), dual-ascent, directed graphs, distributed computing, parallel computing.

I. INTRODUCTION

In this paper, we focus on solving an unconstrained optimization problem of the form

$$\min_{x \in \mathbb{R}^m} \sum_{i=1}^{n} f_i(x),$$

\( (P1) \)

where \( x \) is the decision variable. The computations involved in solving the problem are performed by a group of \( n \) agents which form a communication network. Due to the restrictions imposed by the network, an agent can communicate only with its neighbours in the network. For each \( i \), \( f_i : \mathbb{R}^m \rightarrow \mathbb{R} \) is a convex function privately known to agent \( i \). The goal of each agent is to solve \( (P1) \) in cooperation with its neighbours, without explicitly revealing its function \( f_i \). This restriction on the exchange of \( f_i \)’s could arise due to privacy reasons or due to the communication cost involved in such an exchange. Since each agent minimizes the sum of all functions, it computes a “globally optimal” solution without revealing its own function to other agents. Such distributed optimization problems are encountered in several applications such as achieving average consensus in a distributed manner, distributed power system control, formation control of robots, statistical learning etc. [16].

Distributed optimization problems have been widely studied in the literature. Existing methods for solving these problems can be broadly classified into two types: gradient-descent-based primal methods [10], [15], [17] and Lagrangian-based dual-ascent methods [7], [11], [12], [20]. While most of the methods in the literature assume that the communication network is undirected [11], [12], [20], i.e., communication links between agents are bidirectional; in practice, the network is often directed, i.e., the links are unidirectional. This can occur, for example, if two agents have different broadcast ranges. Generalizing distributed optimization algorithms, which work for undirected graphs, to directed graphs is not straightforward, and in the past has required the introduction of some novel concepts. For example, gradient-descent methods were generalized to directed networks by introducing new concepts such as balancing weights [10], push-sum [15], push-DiGing (Distributed Inexact Gradient tracking) [17] etc. On the other hand, there has been less progress in generalizing Lagrangian-based methods to directed graphs.

A special type of Lagrangian-based dual-ascent method is the Alternating Direction Method of Multipliers (ADMM). Compared to the standard dual-ascent method, ADMM is more robust and amenable to parallel implementation in a multi-agent setting [1]. While, distributed implementation of ADMM is well-studied for undirected graphs [11], [20], an extension to directed graphs was non-existent in the literature (see [17]), until recently. In this paper, we focus on two novel methods for distributed ADMM over directed graphs which have been proposed in the past few years: DC-DistADMM (Directed Constrained Distributed ADMM) in [7], and D-ADMM-FTERC (Distributed Alternating Direction Method of Multipliers using Finite-Time Exact Ratio Consensus) in [4].

The key differences between our method (BW-DADMM), and DC-DistADMM [7] and D-ADMM-FTERC [4] are as follows.

- At each iteration of the respective algorithms, DC-DistADMM requires that, for a fixed \( \epsilon > 0 \), the agents reach consensus up to an \( \epsilon \)-accuracy, while D-ADMM-FTERC requires that the agents reach exact consensus. As noted earlier, ADMM is known to be robust to errors in implementation [1]. For this reason, we do not enforce any
accuracy on the consensus step in BW-DADMM. Instead, we require the agents to communicate with each other \( B \geq 1 \) times per iteration. The value of \( B \) is chosen depending on the parameters of the optimization problem. However, it is observed in simulations that even with \( B = 1 \), the algorithm converges with fewer communication rounds and lesser CPU time compared to that of DC-DistADMM and D-ADMM-FTERC.

- Instead of the push-sum and finite-time ratio-consensus methods, we use the method of balancing weights to achieve consensus among the agents in BW-DADMM. The balancing weights are updated dynamically in a distributed manner. While the operations involved in push-sum and ratio-consensus methods involve divisions, which may cause numerical issues, the balancing weights method involves only addition and multiplication (linear updates), and hence does not suffer from these problems.

- Inspired by the proofs in [12], [17], we prove convergence of BW-DADMM using the small gain theorem. This technique is quite different to the ones used in [7] and [4], and may be of interest independently.

- The problem setup considered in [7] allows for linear constraints on the primal variable, while we only consider unconstrained problems. In that sense, the setup of [7] is more general.

There are also other distributed ADMM algorithms in the literature, some which we summarize here. [3] proposes a distributed ADMM algorithm for undirected graphs while assuming that the weight matrix required for achieving consensus is known to the agents apriori. [9] proposes a Newton method based on the second-order derivative information of the objective function. [22] allows the use of non-smooth objective functions through the use of proximal mapping. However, the paper only guarantees a sublinear convergence rate. [18] proposes an asynchronous distributed ADMM algorithm over a directed network of communication channels with limited bandwidth, again guaranteeing only a sublinear convergence rate. [5] extends the algorithm DADMM-FTERC proposed in [4] to asynchronous updates among the agents. [24] extends the algorithm proposed in an earlier version of this paper ([19]) by allowing an inexact minimization step at each iteration and still guaranteeing convergence of the algorithm. Refer to [23] for a detailed survey of distributed ADMM algorithms.

Our main contributions to the literature are as follows.

- We propose an ADMM algorithm (Algorithm 1: BW-DADMM) which solves the optimization problem \((P1)\) in a distributed manner over directed networks. At each iteration, the algorithm runs an “inner loop” where each agent computes an approximate average of the primal variable iterates of all agents in a distributed manner. This approximate consensus step exploits the inherent robustness of ADMM. This is the key idea that results in a better performance of Algorithm 1 in comparison to the contemporary algorithms in [4], [7].

- Under the assumption that each \( f_i \) in \((P1)\) is strongly convex and smooth, we show that the primal-dual iterates of BW-DADMM converge to their unique optimal points at a geometric rate, provided an explicit bound on the penalty parameter of the ADMM algorithm is satisfied (Theorem 1).

- Through numerical examples, we compare the performance of BW-DADMM to state-of-the-art ADMM methods over directed graphs, in particular [7] and [4]. We specifically demonstrate that the construct of BW-DADMM ensures superior performance for randomly generated wireless sensor networks and sparse graphs, which makes this a conducive choice for real-world networks. Additionally, we demonstrate that BW-DADMM is robust to changes in its parameters, namely, the step-size of the algorithm and the number of communication rounds per iteration.

The rest of the paper is organized as follows. In Section II, we propose a problem equivalent to \((P1)\) that is amenable to distributed implementation. In Section III, we present our distributed ADMM algorithm, BW-DADMM, for solving this equivalent problem. In Section IV, we state our main result which guarantees convergence of the algorithm. We also give an outline of the proof of this result. In Section V, we present numerical examples to compare the performance of our algorithm with other ADMM algorithms over directed graphs. Finally, we give concluding remarks in Section VI. Proofs of all the results can be found in the appendix.

**Notation:** Let \( \log \) denote logarithm with base 10. \( I_n \) be the vector of all ones in \( \mathbb{R}^n \). For a vector \( x \in \mathbb{R}^n \), \( x_i \) be the \( i \)th element of \( x \), \( x^T \) be the transpose of \( x \) and \( \| x \| \) denote the 2-norm of \( x \).

For a matrix \( X \in \mathbb{R}^{n \times m} \), let \( x_i \in \mathbb{R}^m \) denote its \( i \)th row. Let \( \| X \| \) be its Frobenius norm and \( \| X \|_2 \) be its induced 2-norm. Given two matrices \( X \) and \( Y \), let \( \langle X, Y \rangle := Tr(X^TY) \) be their Frobenius inner-product. Given a vector \( x \in \mathbb{R}^n \), let \( diag(x) \in \mathbb{R}^{n \times n} \) be the diagonal matrix with elements \( x_1, \ldots, x_n \) on the diagonal.

For a function \( f : \mathbb{R}^{n \times m} \rightarrow \mathbb{R} \), let \( f^* : \mathbb{R}^{n \times m} \rightarrow \mathbb{R} \) denote the convex conjugate of \( f \) defined as \( f^*(A) = \sup_{X \in \mathbb{R}^{n \times m}} \langle A, X \rangle - f(X) \) for all \( A \in \mathbb{R}^{n \times m} \) such that the supremum is finite. Let \( \nabla f(X) \in \mathbb{R}^{n \times m} \) denote the gradient of \( f \) at \( X \) defined as \( [\nabla f(X)]_{ij} = \partial f(X)/\partial X_{ij} \).

We use the shorthand \( \{ S^k \}_{k \geq 0} \) to denote the sequence \( \{ S^k \} \).
where \( f_i : \mathbb{R}^m \to \mathbb{R} \) is the convex function known only to agent \( i \), and \( x_i \in \mathbb{R}^m \) is the decision variable of agent \( i \). The constraint \( x_i = x_j \) for all \( i, j \in V \) is called the consensus constraint. To write (P2) in a compact form, let \( X = [x_1 \ldots x_n]^T \in \mathbb{R}^{n \times m} \) be the matrix of all decision variables. Then, the consensus constraint can be written as

\[
X = (1_n 1_n^T/n) X.
\]

Further, let \( f(X) = \sum_{i=1}^n f_i(x_i) \). Then, (P2) can be written as

\[
\min_{X \in \mathbb{R}^{n \times m}} f(X)
\]

s.t. \( X = (1_n 1_n^T/n) X \). \hspace{1cm} (P3)

We assume that (P3) is solvable, i.e., there exists an optimal point \( X^* \) of (P3). We make the following assumptions on the function \( f \). These assumptions are standard in the literature whenever a geometric rate of convergence is desired, e.g., [11], [12], [17].

Assumption 2: For each \( i \in \{1, \ldots, n\} \) the function \( f_i \) is \( \mu \)-strongly convex, i.e., \( \exists \mu > 0 \) such that for all \( x, y \in \mathbb{R}^m \),

\[
f_i(y) \geq f_i(x) + \nabla f_i(x)^T(y_i - x_i) + \frac{\mu}{2} ||y_i - x_i||^2.
\]

Further, for each \( i \in \{1, \ldots, n\} \), \( f_i \) is differentiable and \( \nabla f_i \) is \( L \)-Lipschitz, i.e., \( \exists L > 0 \) such that for all \( x, y \in \mathbb{R}^m \),

\[
||\nabla f_i(x) - \nabla f_i(y)|| \leq L ||x_i - y_i||.
\]

An immediate consequence of Assumption 2 is that \( f \) is \( \mu \)-strongly convex and \( \nabla f \) is \( L \)-Lipschitz.

The next two steps ensure that our ADMM algorithm to solve (P3) can be executed in a distributed and parallel manner over the directed graph \( G \). First, we introduce a new variable \( Y \in \mathbb{R}^{n \times m} \) and write (P3) as

\[
\min_{X,Y \in \mathbb{R}^{n \times m}} f(X)
\]

s.t. \( X = Y, \ Y = (1_n 1_n^T/n) Y \). \hspace{1cm} (P4)

This decouples the decision variables \( x_1, \ldots, x_n \) of the agents from each other. Second, we move the constraint \( Y = (1_n 1_n^T/n) Y \) into the objective function using an indicator function as follows. Define

\[
I(Y) = \begin{cases} 
0 & \text{if } Y = (1_n 1_n^T/n) Y, \\
\infty & \text{otherwise} 
\end{cases}
\]

Now, (P4) is equivalent to

\[
\min_{X,Y \in \mathbb{R}^{n \times m}} f(X) + I(Y)
\]

s.t. \( X = Y \). \hspace{1cm} (P5)

The formulation above enables us to impose the constraint \( Y = (1_n 1_n^T/n) Y \) explicitly at each step of the ADMM algorithm, as we shall see later. Since \( f \) is strongly convex under Assumption 2, (P5) has a unique optimal point \( X^* = Y^* \) which satisfies \( (1_n 1_n^T/n) X^* = Y^* \).

To solve (P5) using ADMM, we define the augmented Lagrangian of the problem as

\[
L_\mu(X, Y, A) = f(X) + I(Y) + \langle A, X - Y \rangle + \frac{\rho}{2} ||X - Y||^2,
\]

where \( A \in \mathbb{R}^{n \times m} \) is the dual variable associated with the constraint \( X = Y \) and \( \rho > 0 \) is the penalty parameter. The term \( (\rho/2)||X - Y||^2 \) is called the penalty term. Let \( A^* \) be a dual optimal point of (P5). In the next section, we propose our distributed ADMM algorithm which generates sequences \( \{A_k^*\} \) and \( \{X^*_k\} \) of primal-dual iterates which converge to \( A^* \) and \( X^* \) respectively, at a geometric rate.

III. ALGORITHM

We begin by writing the standard 2-block ADMM algorithm (see [11]) for (P5) as follows. For simplicity, all iterates are initialized at zero. For each \( k \geq 0 \),

\[
X^{k+1} = \argmin_{X \in \mathbb{R}^{n \times m}} L_\mu(X, Y^k, A^k), \hspace{1cm} (2)
\]

\[
Y^{k+1} = \argmin_{Y \in \mathbb{R}^{n \times m}} L_\mu(X^{k+1}, Y, A^k), \hspace{1cm} (3)
\]

\[
A^{k+1} = A^k + \rho(X^{k+1} - Y^{k+1}). \hspace{1cm} (4)
\]

In the algorithm above, the primal iterate \( (X^k, Y^k) \) is divided in two blocks, \( X^k \) and \( Y^k \), which are updated sequentially, before updating the dual iterate \( A^k \). Note that the second block of primal iterates is updated using the latest available value of the first block of primal iterates.

We assume that for all \( k \geq 0 \), each agent \( i \) maintains the set of iterates \( \{x_i^k, y_i^k, a_i^k\} \). To implement the algorithm above in a distributed manner, each agent must update its iterates using only the information received from its in-neighbours. We analyze each update step above to see if such a distributed implementation is possible.

The \( X \) update step: Substituting for \( L_\mu \) from (1) in (2), we obtain

\[
X^{k+1} = \argmin_{X \in \mathbb{R}^{n \times m}} f(X) + \langle A^k, X \rangle + \frac{\rho}{2} ||X - Y^k||^2.
\]

Note that, by definition, \( f \) can be decomposed as \( f(X) = \sum_{i=1}^n f_i(x_i) \). Hence, the update step above is equivalent to

\[
x_i^{k+1} = \argmin_{x_i \in \mathbb{R}^m} f_i(x_i) + \langle a_i^k, x_i \rangle + \frac{\rho}{2} ||x_i - y_i^k||^2
\]

for all \( i \in V \). Each agent \( i \) can independently implement the update step above by solving an unconstrained minimization problem using its local information such as the gradient of the function \( f_i \) and the iterates \( \{a_i^k, y_i^k\} \). The \( X \) update step can thus be implemented parallelly.

The \( Y \) update step: We can derive an explicit expression for \( Y^{k+1} \) in (3) as shown by the following result. (See [7, Appendix B] for a proof of the result.)

Lemma 1: The \( Y \) update step given in (3) is equivalent to

\[
y_i^{k+1} = \frac{1}{n} \sum_{j=1}^n \left( x_j^{k+1} + \frac{a_j^k}{\rho} \right) \hspace{1cm} \text{for all } i \in V.
\]

Note that (7) involves computing an average for each iteration. We use the idea of balancing weights [10], along with a dynamic average consensus type update rule [25] to compute an estimate of \( y_i^{k+1} \) in a distributed manner. This has been proposed earlier for primal gradient tracking, e.g., [10], [17]. We use these techniques to track the average of the dual variable.
Our update rule to estimate \( y_i^{k+1} \) is as follows. Consider an agent \( i \in V \) and an iteration \( k \geq 0 \). Let the agent’s estimate of \( y_i^k \) be denoted by \( \zeta_i^{k+1}(\cdot) \). This estimate is updated dynamically. Let the estimate be initialized as \( \zeta_i^{k+1}(0) = y_i^k + x_i^{k+1} - x_i^k + (a_i^k - a_i^{k-1})/\rho \) for all \( k \geq 0 \).\(^1\) The intuition behind this initialization is to perturb \( y_i^k \), the last-known estimate of the average, by \( x_i^{k+1} - x_i^k + (a_i^k - a_i^{k-1})/\rho \), the change in agent \( i \)'s contribution to the average. Such an initialization is typical of a dynamic average consensus type update rule [8]. Note that this initialization implicitly assumes that agent \( i \) knows the exact value of \( y_i^k \), which may not be possible. This assumption can be relaxed with the following observation. From (4), we know that \( x_i^k - y_i^k = (a_i^k - a_i^{k-1})/\rho \). Hence, \( \zeta_i^{k+1}(0) = x_i^{k+1} \). Thus, initializing the estimate \( \zeta_i^{k+1}(\cdot) \) only requires the knowledge of \( x_i^{k+1} \), which the agent has from the update performed in (6).

Now, agent \( i \) updates its estimate as follows. Consider an integer \( B \geq 1 \), which must satisfy a lower-bound to be specified later. In each iteration \( k \geq 0 \), the agents run an “inner loop” \( B \) number of times where they communicate and update their estimate \( \zeta_i^{k+1}(\cdot) \). Specifically, for all \( k \geq 0 \) and \( b \in \{0, \ldots, B - 1\} \),

\[
\zeta_i^{k+1}(b + 1) = \zeta_i^{k+1}(b) \left(1 - d_i^{\text{out}} w_i^{k+1}(b)\right)
+ \sum_{j \in V : (i,j) \in E} w_j^{k+1}(b) \delta^{k+1}(b), \quad (8)
\]

where \( d_i^{\text{out}} \) is the out-degree of agent \( i \), and \( w_i^{k+1}(b) \in \mathbb{R} \) is the weight used by agent \( j \) to scale its outgoing information. Thus, agent \( i \)'s estimate is computed iteratively by taking a weighted average of its own estimate with those of its in-neighbours. To ensure that the agents reach consensus when the underlying graph is directed, the weights must be chosen appropriately. One such set of weights are the “balancing weights” of a graph. We next describe our update rule used to compute the weights in (8), as introduced in [10].

Let \( D^{\text{out}} := \text{diag}(d_1^{\text{out}}, \ldots, d_n^{\text{out}}) \) be the diagonal matrix of out-degrees of the agents. Let \( C \in \mathbb{R}^{n \times n} \) be the adjacency matrix of the graph, i.e., \( C_{ij} = 1 \) if \( (i,j) \in E \) and \( C_{ij} = 0 \) otherwise. Given a vector \( w \in \mathbb{R}^n \) of weights, let \( W = I - (D^{\text{out}} - C) \text{diag}(w) \) be the weight matrix associated with \( w \).

**Definition 1: (Balancing weights)** Given a graph \( G \), a vector \( w \) is said to be a vector of balancing weights for \( G \) if the weight matrix \( W = I - (D^{\text{out}} - C) \text{diag}(w) \) associated with \( w \) is doubly-stochastic, i.e., \( W1 = 1 \) and \( WT1 = 1 \).

For a balanced graph, i.e., a graph where each node has the same in-degree and out-degree, the vector \( w = 1 \) is trivially the set of balancing weights of the graph. A directed graph is not necessarily balanced and hence computing its vector of balancing weights is non-trivial.

We use the same update rule as described in [10] to compute the set of weights required in (8). Let \( d_\text{max} := \max_{i \in V} d_i^{\text{out}} \) be the maximum out-degree of the graph and \( D \) be the diameter of the graph. Following [10], we initialize the weights as \( w_i^1(0) \leq (1/d_\text{max})^2D + 1 \). Then, for all \( k \geq 0 \) and \( b \in \{0, \ldots, B - 1\} \),

\[
\bar{w}_i^{k+1}(b + 1) = \frac{1}{2} \left( w_i^{k+1}(b) + \frac{1}{d_i^{\text{out}}} \sum_{j \in V : (i,j) \in E} w_j^{k+1}(b) \right),
\]

and \( \bar{w}_i^{k+2}(0) = w_i^{k+1}(1) \). Thus, agent \( i \)'s weight is updated by taking a weighted average of its own weight with those of its in-neighbours. Hence, (9) can be implemented in a distributed manner. Note that for each iterate \( b \in \{0, \ldots, B - 1\} \), the update rule above is implemented sequentially after implementing (8).

The update rule in (9) can be written compactly as

\[
w_i^{k+1}(b + 1) = P w_i^{k+1}(b),
\]

and \( w_i^{k+2}(0) = w_i^{k+1}(B) \), where \( P := (I + (D^{\text{out}} - C^T)/2 \). Further, the update rule in (8) can be written compactly as

\[
\zeta_i^{k+1}(b + 1) = W^{k+1}(b) \zeta_i^{k+1}(b),
\]

where \( (C \odot \delta)((B) \odot \delta)((C)) = \left[ \delta^k(B) \ldots \delta^k(B) \right]^T \) and \( W^{k+1}(b) := I - (D^{\text{out}} - C) \text{diag}(w^{k+1}(b)) \) is the weight matrix associated with the weights \( w^{k+1}(b) \). It is known that the weights updated according to (10) converge to the set of balancing weights of the graph [10]. Additionally, we require that the weight matrix \( W^k(b) \) satisfies certain properties. These properties are stated in our next result.

**Lemma 2:** Let \( W^k(b) = I - (D^{\text{out}} - C) \text{diag}(w^k(b)) \) be the weight matrix associated with the weights \( w^k(b) \) updated as per (10). The matrix has the following properties.

1. For all \( k \geq 1 \), for all \( b \in \{0, \ldots, B - 1\} \), \( \exists p^k(b) \in \mathbb{R}^n \) such that \( p^k(b)1 = 1 \), \( 1^T W^k(b) = 1^T \) and \( W^k(b)p^k(b) = p^k(b) \).
2. \( \exists k \geq 0 \), \( \exists \delta \in [0, 1) \) such that \( \|W^k(b) - p^k(b)1^T\| \leq \delta \) for all \( k \geq \delta \), for all \( b \in \{0, \ldots, B - 1\} \).
3. For all \( b \in \{0, \ldots, B - 1\} \), \( \lim_{k \to \infty} k^\delta(b) = 1/n \).
4. \( \exists k \geq 0 \), \( \exists M \geq 0 \) such that \( \|W^k(b)\| \leq M \) for all \( k \geq \delta \), for all \( b \in \{0, \ldots, B - 1\} \).

The proof of Lemma 2 can be found in the appendix. In essence, Lemma 2 states that \( W^k(b) \) is left-stochastic, while its right eigenvector associated with the eigenvalue 1 converges to \( 1/n \) as \( k \to \infty \). Further, after a large-enough \( k \), \( W^k(b) \) has a bounded norm and \( \delta \)-close to the matrix \( p^k(b)1^T \) in the 2-norm. A version of Fact 2) above appears in [10, Lemma 3], for the element-wise matrix norm instead of the induced 2-norm.

Finally, we look at the distributed implementation of the dual update step.

**The A update step:** From (4), we have \( a_i^{k+1} = a_i^k + \rho(x_i^{k+1} - y_i^{k+1}) \). Thus, each agent \( i \in V \) can compute \( a_i^{k+1} \) independently using its set of iterates \( \{x_i^{k+1}, y_i^{k+1}\} \).

**Remark 1:** The agents need to communicate with their neighbours only during the \( Y \) update step.

The steps implemented by each agent \( i \in V \) are summarized in Algorithm 1.

**Remark 2:** In Algorithm 1, for the ease of notation, we relabel the final estimate \( \zeta_i^{k+1}(B) \) of \( y_i^{k+1} \) to \( y_i^{k+1} \) itself. Thus, henceforth, \( Y^k \) refers to the estimate \( \zeta_i(B) \) of \( Y^k \). This is not
Algorithm 1: BW-DADMM: Balancing Weights Directed ADMM.

Initialize $x_0^k = y_0^k = a_0^k = 0, w_1^k(0) \leq (1/\delta_{\text{out}})^{2D+1}$.

for $k = 0, 1, \ldots$ do

Compute $x_{k+1}^b$ using (6).

end for

Initialize $\zeta_{k+1}^b(0) = x_{k+1}^b$.

for $b = 0, \ldots, B-1$ do

Send $w_{k+1}^b$ and $\zeta_{k+1}^b(b)$ to out-neighbours.

end for

Fix $y_{k+1}^b = \zeta_{k+1}^b(b)$.

Fix $w_{k+1}^{b+1}(0) = w_{k+1}^b(B)$.

Compute $a_{k+1}^b = a_k^b + \rho(x_{k+1}^b - y_{k+1}^b)$.

end for

to be confused with the exact value of $Y^k$ defined by (7), which will not be referred to in the rest of the paper.

Remark 3: In proving convergence of BW-DADMM, we use the fact that

$$1^T Y^k = 1^T X^k, \quad 1^T A^k = 0$$

for all $k \geq 0$. It can be verified that this is indeed true if we initialize the iterates such that $1^T Y^0 = 1^T X^0$ and $1^T A^0 = 0$, and use the fact that $1^T W^k(b) = 1^T$ (Lemma 2).

Next, we present our main result which states that if the parameters of Algorithm 1 are chosen appropriately, then its primal-dual iterates converge to their respective optimal points with a geometric rate.

IV. MAIN RESULT

In this section, we state our main result and then give an outline of its proof. The details of the proof can be found in the appendix.

Theorem 1: Given a problem of the form (P5), suppose that the parameters of BW-DADMM (Algorithm 1) are chosen such that the number of communication rounds per iteration, $B$, satisfies

$$B \geq \max \left\{ 1, \left[ \frac{\log(\gamma_1 \gamma_3)}{\log(1/\delta_1)} \right] \right\}$$

the convergence rate $\lambda \in (0, 1)$ satisfies

$$\frac{L}{\mu} \left( \frac{1}{\lambda^2} - 1 \right) < \left( \frac{\lambda}{1 + \lambda} \right) \min \left\{ \frac{1}{2}, \hat{c}_3 \right\}$$

and the penalty parameter $\rho$ satisfies

$$\rho \in \left( \frac{1}{L \left( \frac{1}{\lambda^2} - 1 \right)}, \left( \frac{\lambda \mu}{1 + \lambda} \right) \min \left\{ \frac{1}{2}, \hat{c}_3 \right\} \right),$$

where

$$\gamma_1 := 1/\mu,$$

$$\gamma_3 := \hat{c}_1 + \rho \left( 1 + \frac{2\hat{c}_1}{\mu} \right) \left( \frac{\mu(1 + \lambda)}{\lambda \mu - 2\rho(1 + \lambda)} \right),$$

for some non-negative constants $c_0, c_\lambda$, where $(X^*, A^*)$ is the unique primal-dual optimal point of (P5).

We make the following remarks on the result.

Remark 4: It is easy to check that a (large-enough) $\lambda \in (0, 1)$ which satisfies (13) always exists. As $\lambda \rightarrow 1$, the LHS of (13) converges to zero, while the RHS converges to a positive number. This implies that a $\rho > 0$ satisfying (14) always exists.

Remark 5: If $f$ is $\mu$-strongly convex and $L$-smooth (as is the case due to Assumption 2), then $\kappa = L/\mu \geq 1$ is called the condition number of $f$. We can argue that the convergence of BW-DADMM is faster if the problem is well-conditioned, i.e., $\kappa$ is small. To see this, note that with a decrease in $\kappa = L/\mu$, the LHS of (13) decreases, while the RHS increases (since $\hat{c}_3$ increases). Thus, smaller values of $\kappa \geq 1$ give a wider range of values of the convergence rate $\lambda$ that satisfy (13).

Remark 6: At each iteration of BW-DADMM, the agents solve the unconstrained optimization problem (6). In general, this problem may not have a closed-form solution. However, since the problem is convex, it can be solved numerically. Such a numerical solution may be approximate at best. Analyzing the convergence of BW-DADMM when the agents can only find an approximate solution to (6) at each iteration is beyond the scope of this paper. It is known that, in general, ADMM converges in such a scenario [2].

Remark 7: From (12), we observe that to guarantee convergence of BW-DADMM, the required number of communication rounds per iteration, $B$, is small if $\delta < 1$, as defined in Lemma 2, is small. Note that $\delta$ represents the convergence rate to average consensus [21]. Thus, if the network is capable of achieving consensus faster (smaller $\delta$), then the value of $B$ is smaller. $\delta$ depends on the communication network structure, and hence, in many distributed paradigms, is not known to the agents. For practical purposes, one can choose $\delta \in (0, 1)$ very close to 1.

The proof of Theorem 1 uses the small gain theorem. This proof is inspired by the ideas in [17] and [12]. We briefly recall the theorem after introducing some notation. Given a sequence $\{S^k\}$ of matrices in $\mathbb{R}^{n \times m}$ and a convergence rate $\lambda \in (0, 1)$,
let
\[
\|S\|_K^\lambda = \max_{k = 0, \ldots, K} \lambda^k \frac{\|S_k\|}{\|X_k\|}, \quad |S|^\lambda = \sup_{k \geq 0} \frac{\|S_k\|}{\|X_k\|} \tag{15}
\]
for all \(K \geq 0\). Further, given two sequences \(\{S_k^1\}, \{S_k^2\}\) and a constant \(\gamma \geq 0\), we use the notation \(S_1 \rightarrow S_2\) to denote \(\|S_2\|^{\gamma, K} \leq \gamma \|S_1\|^{\gamma, K} + \omega\) for all \(K \geq 0\), for some constant \(\omega \geq 0\). Now, the small gain theorem can be stated as follows.

**Proposition 1:** [17, Theorem 3.7] (Small gain theorem) Given \(m\) sequences \(\{S_k^1\}, \ldots, \{S_k^m\}\) of vectors in \(\mathbb{R}^n\), suppose there exist non-negative constants \(\gamma_1, \ldots, \gamma_m\) and a convergence rate \(\lambda \in (0, 1)\) satisfying the cycle of relations
\[
S_1 \xrightarrow{\lambda} S_2 \xrightarrow{\lambda} S_3 \xrightarrow{\lambda} \cdots \xrightarrow{\lambda} S_m \xrightarrow{\lambda} S_1
\]
such that \(\gamma_1 \ldots \gamma_m < 1\). Then, there exist constants \(c_j \geq 0\) such that \(\|S_j\|^\lambda \leq c_j\) for all \(j \in \{1, \ldots, m\}\).

Note that, if the conditions of the small gain theorem are satisfied, then \(S_j^k \rightarrow 0\) at a geometric rate of \(\lambda \in (0, 1)\) for all \(j \in \{1, \ldots, m\}\).

Now, we prove Theorem 1 using the small gain theorem as follows. Let
\[
\hat{A}^k := A^k - A^*, \quad \hat{X}^k := X^k - X^*
\]
To show that \(A^k \rightarrow A^*\) and \(X^k \rightarrow X^*\) with a geometric rate of \(\lambda\), it is enough to show that \(|\hat{A}|^\lambda\) and \(|\hat{X}|^\lambda\) are bounded. To show this, we first prove that there exist non-negative numbers \(\gamma_1, \gamma_2, \gamma_3\) such that the cycle of relations
\[
\hat{A} + \rho \Delta Y \xrightarrow{\lambda} X_1 \xrightarrow{\lambda} Y_1 \xrightarrow{\lambda} \hat{A} + \rho \Delta Y \tag{16}
\]
holds with \(\gamma_1 \gamma_2 \gamma_3 < 1\), where \(\Delta Y^k := Y^k - Y^{k-1}\), \(X^k := X^k - \left(1 - \frac{T^k}{n}\right)X^k\), \(Y^k := Y^k - \left(1 - \frac{T^k}{n}\right)Y^k\) for all \(k \geq 0\). The proof of each arrow in (16) is provided in the appendix.

In particular, we show that the conditions given in Theorem 1 are sufficient to prove the cycle of relations in (16) such that \(\gamma_1 \gamma_2 \gamma_3 < 1\). Then, by the small gain theorem (Proposition 1), it follows that \(|\hat{Y}|^\lambda\) and \(|\hat{A} + \rho \Delta Y|^\lambda\) are bounded.

After proving that \(|\hat{Y}|^\lambda\) and \(|\hat{A} + \rho \delta Y|^\lambda\) are bounded, it is shown in the appendix that, for some constants \(\gamma_0 \geq 0, \gamma_x \geq 0, \omega_1, \omega_x\),
\[
|\hat{A}|^\lambda \leq \gamma_0 |\hat{Y}|^\lambda + \omega,
\]
\[
|\hat{X}|^\lambda \leq \gamma_0 |\hat{A} + \rho \delta Y|^\lambda + \omega_x. \tag{17}
\]
Then, it follows that \(|\hat{A}|^\lambda \leq c_0\) and \(|\hat{X}|^\lambda \leq c_x\) for some non-negative constants \(c_0\) and \(c_x\). From the definition of \(|S|^\lambda\) given in (15), this implies \(|\hat{A}|^\lambda \leq c_0 \lambda^k\) and \(|\hat{X}|^\lambda \leq c_x \lambda^k\) for all \(k \geq 0\), which completes the proof of Theorem 1.

Next, we present numerical examples to illustrate our main result, and to compare BW-DADMM with other ADMM algorithms over directed graphs.

V. NUMERICAL EXAMPLES

The observations noted in this section were made across many examples. For simplicity, we present the results from one such example.²

²The datasets generated during the current study are available from the corresponding author on request.

A. SENSOR NETWORK

**Problem setup:** We consider a sensor network of \(n = 50\) agents generated as follows. First, each agent \(i \in V\) is placed at a location \(l_i \in \mathbb{R}^2\) in the unit square \([0, 1]^2\) uniformly at random. Then, each agent \(i \in V\) is assigned a broadcast range \(r_i\) generated uniformly at random between 0.2 and 0.4. Then, a directed network is established as follows. If agent \(i\) is within the broadcast range of agent \(j\), then we create an edge \((i, j)\) in the graph so that agent \(j\) can send information to agent \(i\). Mathematically, \((i, j) \in E\) if and only if
\[
\|l_i - l_j\| \leq r_j. \tag{18}
\]

We consider the problem of sensor fusion, where the goal of each agent is to estimate a common parameter of interest using data from the neighbouring agents. The local objective function of agent \(i\) is \(f_i(x) = (1/2)\|H_i x - g_i\|^2\), where each entry of the measurement matrix \(H_i \in \mathbb{R}^{10 \times 2}\) and the data matrix \(g_i \in \mathbb{R}^2\) is generated independently at random from a standard normal distribution. To estimate the parameter, the agents must solve problem (P1) in a distributed manner.

**Algorithms chosen for comparison:** We consider DC-DistADMM [7] and D-ADMM-FTERC [4]. Note that the bounds on the parameters of each of these algorithms, such as the ones on \(\rho\) and \(B\) given in Theorem 1, are often very conservative. Hence, for each of these algorithms, we perform a numerical grid search over a subset of parameter values to find the parameter values that give the best convergence rate for the problem under consideration. Note that this method of grid search has earlier been suggested in [17].

**Convergence rate:** A plot of the normalized primal residual obtained by each of the algorithms mentioned above is shown in Fig. 1. It can be observed that, in this example, BW-DADMM performs better than the other two algorithms in terms of the number of communication rounds and CPU time required for convergence, whereas DC-DistADMM performs better in terms of the number of iterations needed to converge. Fig. 1 (top) also illustrates a geometric upper-bound on the convergence rate of BW-DADMM.

It is worth noting that the algorithms mentioned above also generate a sequence \(\{A^k\}\) of dual variable iterates which converge to the dual optimal point \(A^*\) (not shown here due to space constraints). This is not the case with gradient-based algorithms.

**Communication cost:** For each algorithm, the number of values sent by an agent per communication round is \(O(m)\), where \(m\) is the dimension of the unknown parameter.

B. Graphs With Large Diameters

Most of the graphs observed in practice are sparse and hence have large diameters. It was observed that BW-DADMM performs much well on graphs with large diameters, such as a directed circle graph and an undirected line graph. The residuals generated by each algorithm on a directed circle graph are often very conservative. Hence, for each of these algorithms, we perform a numerical grid search over a subset of parameter values to find the parameter values that give the best convergence rate for the problem under consideration. Note that this method of grid search has earlier been suggested in [17].
by $\eta(W)^k$, where $\eta(W)$ is the second-smallest eigenvalue of the weight matrix $W$ used to achieve consensus [21]. Further, $\eta(W)$ has a lower-bound that is inversely proportional to the diameter of the graph [14]. Thus, graphs with larger diameters take more iterations to achieve consensus. BW-DADMM does not require any accuracy on the inner consensus loop, while the other algorithms require either exact consensus or $\epsilon$-consensus at each step. ADMM appears to be robust to errors in consensus.

Hence, enforcing consensus accuracy at each iteration may be an overkill.

C. Robustness of BW-DADMM to Changes in Its Parameters

Robustness to changes in $B$: We study the effect of changing the number of communication rounds per iteration ($B$) in the inner loop of our ADMM algorithm. As observed in Fig. 3 (top), with an increase in $B$, the number of iterations required to reach the optimal value decreases, while there is no change in the final value.

Robustness to changes in $\rho$: We observe in Fig. 3 (bottom), that the primal residual decreases monotonically for a large range of values of $\rho$, albeit with different rates. On the other hand, primal-descent algorithms are known to be sensitive to their step-size.

VI. Conclusion

We proposed an ADMM algorithm called BW-DADMM to solve distributed optimization problems over directed graphs. The algorithm uses the ideas of balancing weights and dynamic average consensus. Under the assumption that the objective function is strongly convex and smooth, we showed that the primal-dual iterates of the algorithm converge to their unique optimal points at a geometric rate, provided the parameters of the algorithm are chosen appropriately. Through a numerical example, we demonstrated that BW-DADMM gives a comparable performance to some state-of-the-art ADMM methods over directed graphs. Additionally, the algorithm was observed to be robust to changes in its parameters. In the future, it will be interesting to see if convergence of the algorithm can be
guaranteed by relaxing the assumptions of strong convexity and smoothness. Further, it will be interesting to extend the algorithm to time-varying graphs.

APPENDIX

PROOF OF Lemma 2

1) The statement follows from the fact that, by definition, $W_k^T(b) = I - (D^{out} - A) \text{diag}(w_k(b))$ is column stochastic.

2) It is shown in [10, Lemma 1] that $\lim_{k \to \infty} p_k$ exists. Hence, $\lim_{k \to \infty} W_k^T(b) = p_k^{(k-1)(B^{-1})}w_k(0)$ exists and is independent of $b$. This implies $\lim_{k \to \infty} W_k(b)$ exists and is independent of $b$. Let $W^\infty := \lim_{k \to \infty} W_k(b)$. Now, by the triangle inequality,

$$\|W_k(b) - p_k(b)1T\| \leq \|W_k(b) - W^\infty\| + \|W^\infty - 11^T/n\| + \|11^T/n - p_k(b)1T\|.$$  

(19)

We analyze each term in the RHS of (19) one-by-one.

a) $\|W_k(b) - W^\infty\|$: By definition of $W^\infty$, $\|W_k(b) - W^\infty\| \to 0$ as $k \to \infty$.

b) $\|W^\infty - 11^T/n\|$: It is shown in [10, Lemma 1] that $W^\infty$ is a doubly-stochastic matrix. Further, it is shown in [10, Lemma 2] that $W^\infty$ is a non-negative matrix with $[W^\infty]_{ij} > 0$ for all $i \in V$ and $[W^\infty]_{ij} > 0$ if and only if $(i,j) \in E$. Using these facts, we first show that $W^\infty(W^\infty)^T$ is a primitive matrix. To show this, it is enough to show that $(W^\infty(W^\infty)^T)^D$ is a positive matrix, where $D$ is the diameter of $G$. For any $(i,j) \in V \times V$,

$$[(W^\infty)^D]_{ij} = \sum_{i_1 \in V} \sum_{i_2 \in V} \cdots \sum_{i_{D} \in V} [W^\infty]_{i_1i_2} \cdots [W^\infty]_{i_Dj}.$$  

By definition, $D$ is the length of the largest path connecting any two nodes. Moreover, $G$ is strongly connected. Hence, any given $i,j \in V$, there exists a path $(q_1i), (q_2q_1), \ldots, (j, q_D)$ from $i$ to $j$. Hence,

$$[(W^\infty)^D]_{ij} \geq [W^\infty]_{iq_1} [W^\infty]_{q_1q_2} \cdots [W^\infty]_{q_Dj} > 0.$$  

Thus, $(W^\infty)^D$ is a positive matrix. Now, $[W^\infty(W^\infty)^T]_{ij} \geq [W^\infty]_{ij}[W^\infty]_{ji} > 0$. Thus, $(W^\infty(W^\infty)^T)^D$ is also a positive matrix. It then follows that $W^\infty(W^\infty)^T$ is a primitive matrix. This implies $\rho(W^\infty(W^\infty)^T - 11^T/n) < 1$ [13, (8.3.16)]. Now,

$$\|W^\infty - 11^T/n\| \leq \rho\left((W^\infty - 11^T/n)(W^\infty - 11^T/n)^T\right)$$  

$$= \rho\left(W^\infty(W^\infty)^T - 11^T/n\right)$$  

$$< 1,$$  

where we have used the facts $W^\infty 1 = 1, 1^T W^\infty = 1^T$ as shown in [10, Lemma 1]. Thus, $\|W^\infty - 11^T/n\| < 1.$

c) $\|11^T/n - p_k(b)1T\|$: Consider any $\epsilon > 0$. We show that there exists $k_1$ such that $\|11^T/n - p_k(b)1T\| < \epsilon$ for all $k > k_1$. For all $k \geq 0$ and $l \geq 0$,

$$\|11^T/n - p_k(b)1T\| \leq \|11^T/n - (W^\infty)^l\| + \|(W^\infty)^l - (W_k(b))^l\| + \|(W_k(b))^l - p_k(b)1T\|.$$  

(20)

We argued above that $W^\infty$ is a primitive matrix with $1^T W^\infty = 1^T$ and $W^\infty 1 = 1$. Hence, $\rho(W^\infty - 11^T/n) < 1$. It follows that there exists $k_1$ such that $\|(W^\infty)^l - 11^T/n\| < \epsilon/3$ for all $l \geq l_1$ [13, (8.3.10)]. Following the same arguments as with $W^\infty$ above, we can show that $\rho((W^\infty)^l - p_k(b)1T) < 1$ for all $k \geq 0$ and $0 \leq b \leq B - 1$. Hence, for all $k \geq 0$, $\|W^\infty(W^\infty)^T - 11^T/n\| < \epsilon/3$ for all $l \geq l_2$. Moreover, since $\lim_{k \to \infty} W_k(b) = W^\infty$, the eigenvalues of $W_k(b)$ converge to those of $W^\infty$. This implies $l_2$ is independent of $k$ (see [13, (8.3.10)], which implies that the rate of convergence of $p_k(b)1T$ to $p_k(b)1T$ depends only on the second largest eigenvalue $W_k(b)$). Now, let $l_0 := \max(l_1, l_2)$. Note that given any matrix $C \in \mathbb{R}^{n \times n}$, $C^l_{00}$ is a continuous function of the entries of $C$. Hence, $\lim_{k \to \infty} W_k(b) = W^\infty$ implies $\exists k_1 \geq 0$ such that $\|(W^\infty)^l_{00} - (W_k(b))_{00}\| < \epsilon/3$ for all $k \geq k_1$. Substituting the aforementioned facts in (20), we have $\|11^T/n - p_k(b)1T\| < \epsilon$ for all $k \geq k_1$. This implies $\|11^T/n - p_k(b)1T\| \to 0$ as $k \to \infty$.

Using the results derived for each of the three terms above, from (19) we can conclude that $\|W_k(b) - p_k(b)1T\| < 1$ for $k$ large-enough.

(3) The statement follows from the fact that $\|11^T/n - p_k(b)1T\| \to 0$ as $k \to \infty$, which we have proved above.

(4) For all $k \geq 1$, $\|W_k(b)\| < \|W_k(b) - p_k(b)1T\| + \|p_k(b)1T\|$. Now, choosing $\hat{k} \geq \tilde{k}$ implies $\|W_k(b) - p_k(b)1T\| \leq \delta$ for all $k \geq \hat{k}$. On the other hand, $\|p_k(b)1T\|$ is bounded since $p_k(b)$ converges to $1/n$. Thus, $\|W_k(b)\| \leq M$ for all $k \geq \hat{k}$ for some $M \geq 0$.

SOME INTERMEDIATE RESULTS

We state some useful intermediate results.

Lemma 3: Let $\{p_k\}, \{q_k\}, i \in \{1, \ldots, m\}$, be sequences in $\mathbb{R}$ such that $q_k^i \geq 0$ for all $i,k$. Suppose $(p_k)^2 \leq \sum_{i=1}^{m} (q_k^i)^2 + c$ for some constant $c \geq 0$, for all $k$. Then, $p_k \leq \sum_{i=1}^{m} (q_k^i) + \sqrt{c}$ for all $k$.

Proof: Since $\{q_k^i\}$ is a non-negative sequence for all $i \in \{1, \ldots, m\}$ and $c \geq 0$, we have

$$\left(\sum_{i=1}^{m} (q_k^i) + \sqrt{c}\right)^2 \geq \sum_{i=1}^{m} (q_k^i)^2 + c$$

for all $k$. Now, using $(p_k)^2 \leq \sum_{i=1}^{m} (q_k^i)^2 + c$, we obtain

$$\left(\sum_{i=1}^{m} (q_k^i) + \sqrt{c}\right)^2 \geq \sum_{i=1}^{m} (q_k^i)^2 + c \geq (p_k)^2.$$
Let $f^*$ denote the conjugate of $f$, i.e., $f^*(a) = \sup_{x} \langle A, X \rangle - f(X)$ for all $a \in \mathbb{R}^n$ such that the supremum is finite.

**Lemma 4:** The dual optimal point $A^*$ of (P5) satisfies $(1_n1_n^T/n)A^* = 0$.

**Proof:** Given the standard Lagrangian $L_0$ of (P5) (1) with $\rho = 0$, the dual problem of (P5) is given by

$$
\begin{align*}
\sup_A \inf_{(X,Y)} L_0(X,Y,A) \\
= \sup_A \left( \inf_X (f(X) + \langle A, X \rangle) + \inf_Y f(I(Y) - \langle A, Y \rangle) \right) \\
= \sup_A \left( -f^*(-A) + \inf_{Y=1_n1_n^T/n} \langle A, Y \rangle \right) \\
= \sup_{(1_n,1_n^T/n)A=0} -f^*(-A)
\end{align*}
$$

Thus, the dual optimal point satisfies $(1_n1_n^T/n)A^* = 0$. $\blacksquare$

**Proposition 2:** [6, Theorem 6] Under Assumption 2, $f^*$ is $1/L$-strongly convex and $1/\mu$-smooth.

**Lemma 5:** The primal-dual iterates of Algorithm 1 satisfy $X^{k+1} = \nabla f^*(-A^{k+1} - \rho Y^{k+1})$ for all $k \geq 0$ and the primal-dual optimal points of (P5) satisfy $X^* = \nabla f^*(-A^*)$.

**Proof:** From the definition of $f^*$, we have

$$
f^*(-A) = \sup_{x} \langle A, X \rangle - f(X).
$$

This implies

$$
\nabla f^*(-A) = \arg\max_X \langle A, X \rangle - f(X) = \arg\max_X \langle A, X \rangle + f(X).
$$

Now, from the update step (5) of Algorithm 1, we have, for all $k \geq 0$,

$$
\nabla f(X^{k+1}) + A^{k+1} + \rho (X^{k+1} - Y^{k}) = 0.
$$

Using (4), this implies

$$
\nabla f(X^{k+1}) + A^{k+1} + \rho Y^{k+1} = 0.
$$

Hence, we have

$$
X^{k+1} = \arg\min_X \nabla f(X) + \langle A^{k+1} + \rho Y^{k+1}, X \rangle.
$$

Now, using above relation with (21) gives $\nabla f^*(-A^{k+1} - \rho Y^{k+1}) = X^{k+1}$, which is the first part of the lemma. Next, by definition of a primal-dual optimal point, we have

$$
(X^*, Y^*) = \arg\min_{(X,Y)} L_0(X,Y,A^*),
$$

where $L_0$ is the Lagrangian of (P5) (1) with $\rho = 0$. This implies $X^* = \arg\min_X f(X) + \langle A^*, X \rangle$. Using this relation with (21), it follows that $\nabla f^*(-A^*) = X^*$. $\blacksquare$

**Proof of each arrow in (16)**

**Proof of the first arrow $(\bar{A} + \rho \Delta Y \rightarrow X_\perp)$:** From Proposition 2, we obtain that for all $k \geq 0$,

$$
\begin{align*}
\nabla f^*(-A^{k+1} - \rho Y^{k+1}) - \nabla f^*(-A^*) \\
\leq \frac{1}{\mu} ||A^{k+1} - A^* + \rho Y^{k+1}|| = \frac{1}{\mu} ||A^{k+1} + \rho \Delta Y^{k+1}||.
\end{align*}
$$

Using Lemma 5 with the relation above, we have

$$
\begin{align*}
\parallel X^{k+1} - X^* \parallel^2 \leq \frac{1}{\mu} \parallel A^{k+1} + \rho \Delta Y^{k+1} \parallel^2.
\end{align*}
$$

Further, $\parallel X^{k+1} - X^* \parallel^2 = \parallel X^{k+1}_\perp \parallel^2 + \parallel (1_n1_n^T/n)X^{k+1} - X^* \parallel^2$. Substituting this relation in (22), we have

$$
\begin{align*}
\parallel X^{k+1}_\perp \parallel \leq \frac{1}{\mu} \parallel A^{k+1} + \rho \Delta Y^{k+1} \parallel.
\end{align*}
$$

Hence,

$$
\begin{align*}
\parallel X_\perp \parallel \leq \frac{1}{\mu} \parallel \bar{A} + \rho \Delta Y \parallel^L \parallel X_\perp \parallel + \parallel X_\parallel^0 \parallel
\end{align*}
$$

for all $K \geq 0$. Thus, $\gamma_1 := 1/\mu$.

**Proof of the second arrow $(X_\perp \rightarrow Y_\perp)$:** Using the update rule (11) of $\zeta(k)$ and the first property in Lemma 2, we have, for all $k \geq 1$ and $b \in \{0, \ldots, B-1\}$,

$$
\begin{align*}
(I - p^k(b)1^T)^T \zeta(k)(b+1) \\
= (I - p^k(b)1^T)^T W^k(b)^T \zeta(k)(b) \\
= (W^k(b) - p^k(b)1^T)^T \zeta(k)(b) \\
= (W^k(b) - p^k(b)1^T)^T - p^k(b)1^T + p^k(b)1^T) \zeta(k)(b) \\
= (W^k(b) - p^k(b)1^T)^T - W^k(b)p^k(b)1^T \\
\quad + p^k(b)1^T p^k(b)1^T) \zeta(k)(b) \\
= (W^k(b) - p^k(b)1^T) (I - p^k(b)1^T) \zeta(k)(b).
\end{align*}
$$

Hence,

$$
\begin{align*}
(I - p^k(b)1^T)^T + 11^T/n - 11^T/n) \zeta(k)(b+1) \\
= (W^k(b) - p^k(b)1^T) \\
\quad \times (I - p^k(b)1^T) \zeta(k)(b).
\end{align*}
$$

This implies

$$
\begin{align*}
\zeta_k^L(b+1) + (1/n - p^k(b)) 1^T \zeta_k^L(b+1) \\
= (W^k(b) - p^k(b)1^T) \left[ \zeta_k^L(b) + (1/n - p^k(b)) 1^T \zeta_k^L(b) \right];
\end{align*}
$$

where $\zeta_k^L(b) := (I - 11^T/n) \zeta_k^L(b)$. This implies

$$
\begin{align*}
\parallel \zeta_k^L(b+1) + (1/n - p^k(b)) 1^T \zeta_k^L(b+1) \parallel \\
\leq \parallel W^k(b) - p^k(b)1^T \parallel_\infty \parallel \zeta_k^L(b) \parallel + (1/n - p^k(b)) 1^T \zeta_k^L(b),
\end{align*}
$$

Since $\zeta^L(0) = X^k$ and $1^T W^k(b) = 1^T$, we have $1^T \zeta^L(b+1) = 1^T W^k(b) \zeta^L(b) = 1^T X^k$. Further, by the second property in Lemma 2, $\exists k \geq 0, \exists \delta \in [0, 1]$ such that for all $k \geq \delta$, $\parallel W^k(b) - p^k(b)1^T \parallel_\infty \leq \delta$. Hence, for all $k \geq \delta$,

$$
\begin{align*}
\parallel \zeta_k^L(b+1) + (1/n - p^k(b)) 1^T X^k \parallel \\
\leq \delta \parallel \zeta_k^L(b) + (1/n - p^k(b)) 1^T X^k \parallel,
\end{align*}
$$

Now, by the third property in Lemma 2, we know that $\lim_{k \rightarrow \infty} \parallel 1/n - p^k(b) \parallel = 0$. Since $\parallel \cdot \parallel$ is a continuous function, there must exist a $k \geq k$ such that for all $k \geq k, \parallel \zeta_k^L(b+1) \parallel \leq \delta \parallel \zeta_k^L(b) \parallel$. Using this inequality iteratively from $b = 0$ to...
for some constant \( c_1 \geq 0 \). Thus, \( \gamma_2 := \delta_B \).

Proof of the third arrow \((Y_k \xrightarrow{\mathcal{T}_A} \mathcal{A} + \rho \Delta Y)\): We prove the third arrow in two steps. In the first step, we bound \( \Delta Y \) by a linear combination of \( Y_k \) and \( A \). In the second step, we bound \( \mathcal{A} \) by a linear combination of \( Y_k \) and \( \Delta Y \). To complete the proof, we use the two bounds together.

Step 1: Using triangle inequality,
\[
\|\Delta Y^{k+1}\| \\
\leq \|Y^{k+1} - Y^k\| \\
\leq \|Y^{k+1} - (11 T/n) Y^k + Y^k - (11 T/n) Y^k\| \\
+ \|(11 T/n) Y^{k+1} - (11 T/n) Y^k\| \\
= \|Y^{k+1} - Y^k\| + \|Y^k - (11 T/n) Y^k\|.
\]

We know that \(1^T Y^k = 1^T X^k\) for all \( k \geq 0\). Hence,
\[
\|\Delta Y^{k+1}\| \\
\leq \|Y^{k+1}_\perp\| + \|Y^k_\perp\| + \|(11 T/n) X^k - (11 T/n) X^k\| \\
\leq \|Y^{k+1}_\perp\| + \|Y^k_\perp\| + \|X^{k+1}_\perp\| + \|X^k_\perp\| + \|X^{k+1} - X^k\|.
\]

From (23), we know that \(\|X^{k+1}_\perp\| \leq (1/\mu)\|\mathcal{A}^{k+1} + \rho \Delta Y^{k+1}\|\) for all \( k \geq 0\). Using this with Lemma 5 and Proposition 2, we have
\[
\|X^{k+1} - X^k\| \\
\leq \|X^{k+1} - X^k\| + \|X^k - X^k\| \\
= \|\nabla f^*(-A^{k+1} - \rho \Delta Y^{k+1}) - \nabla f^*(-A^k)\| \\
+ \|\nabla f^*(-A^k - \rho \Delta Y^{k}) - \nabla f^*(-A^k)\| \\
\leq \frac{1}{\mu} \left(\|\mathcal{A}^{k+1} - A^* + \rho \Delta Y^{k+1}\| + \|A^k - A^* + \rho \Delta Y^{k}\|\right) \\
\leq \frac{1}{\mu} \left(\|\mathcal{A}^{k+1}\| + \|\mathcal{A}^k\| + \rho (\|\Delta Y^{k+1}\| + \|\Delta Y^{k}\|)\right).
\]

Hence,
\[
\|\Delta Y^{k+1}\| \\
\leq \|Y^{k+1}_\perp\| + \|Y^k_\perp\| + \frac{2}{\mu} \left(\|\mathcal{A}^{k+1}\| + \|\mathcal{A}^k\| + \rho (\|\Delta Y^{k+1}\| + \|\Delta Y^{k}\|)\right).
\]

Dividing by \(\lambda^{k+1}\) on both sides, we have
\[
\frac{\|\Delta Y^{k+1}\|}{\lambda^{k+1}} \leq \frac{\|Y^{k+1}\|}{\lambda^{k+1}} + \frac{1}{\lambda} \left(\frac{\|Y^k\|}{\lambda^k} + \frac{2}{\mu} \left(\frac{\|\mathcal{A}^{k+1}\|}{\lambda^{k+1}} + \frac{1}{\lambda} \frac{\|\mathcal{A}^k\|}{\lambda^k} + \rho \left(\frac{\|\Delta Y^{k+1}\|}{\lambda^{k+1}} + \frac{1}{\lambda} \frac{\|\Delta Y^{k}\|}{\lambda^k}\right)\right)\right).
\]

for all \( k \geq 0\). Taking supremum over \( k = 0, 1, \ldots, K - 1\), we have
\[
\|\Delta Y\|^{\lambda,K} \leq \left(1 + \frac{1}{\lambda}\right) \|Y\|^{\lambda,K} + \frac{2}{\mu} \left(\left(\frac{1 + 1}{\lambda}\right) (\|\mathcal{A}\|^{\lambda,K} + \rho \|\Delta Y\|^{\lambda,K}) + c_2\right).
\]

From (14), we know that \(\lambda \mu - 2\rho(1 + \lambda) > 0\). Hence,
\[
\|\Delta Y\|^{\lambda,K} \leq \left(\frac{\lambda \mu}{\mu(1 + \lambda)}\right) \|Y\|^{\lambda,K} + \frac{2}{\mu} \left(\frac{\mu(1 + \lambda)}{\lambda \mu - 2\rho(1 + \lambda)}\right) \|\mathcal{A}\|^{\lambda,K} + c_2.
\]

Step 2: Let \(\Delta A^{k+1} = A^{k+1} - A^k\). For all \( k \geq 0\),
\[
\|\mathcal{A}^k\|^{2} = \|A^k - A^k - A^k - A^k\|^2 \\
= \|\Delta A^{k+1}\|^2 + \|\mathcal{A}^{k+1}\|^2 \\
+ 2(\mathcal{A}^{k+1} - A^k, A^{k+1} - A^k) \\
\geq \|\mathcal{A}^{k+1}\|^2 + 2(\mathcal{A}^{k+1} - A^k, A^{k+1} - A^k).
\]

This implies
\[
\|\mathcal{A}^{k+1}\|^2 \leq \|\mathcal{A}^k\|^2 + 2(\mathcal{A}^{k+1} - A^k, A^{k+1} - A^k).
\]

We try to bound the term \(2(\mathcal{A}^{k+1} - A^k, A^{k+1} - A^k)\) above. The \(A\) update step in Algorithm 1 can be written as
\[
\mathcal{A}^{k+1} = \arg\min_{A, (1,1 T/n), A=0} \rho(Y^{k+1} - X^{k+1}, A) + \frac{1}{2} \|A - A^k\|^2 \\
= \arg\min_{A, (1,1 T/n), A=0} \rho(Y^{k+1} - X^{k+1}, A) + \frac{1}{2} \|A - A^k\|^2,
\]

From Lemma 4, we know that \((1,1 T/n) A^* = 0\). Evaluating the optimality condition for the minimization problem above at \( A = A^*\), we obtain
\[
\langle \rho(Y^{k+1} - X^{k+1}) + A^{k+1} - A^k, A^* - A^{k+1}\rangle \geq 0.
\]

Using Lemma 5, this implies
\[
\langle A^{k+1} - A^k, A^{k+1} - A^*\rangle \\
\leq \rho\langle \nabla f^*(A^{k+1} - \rho \Delta Y^{k+1}) - Y^{k+1}_\perp, A^{k+1} - A^*\rangle.
\]

For the ease of notation, let \( A = -A^{k+1} - \rho \Delta Y^{k+1} \). Now, we bound the term \(\rho \langle \nabla f^*(A) - Y^{k+1}_\perp, A^{k+1} - A^*\rangle\). Since \(f^*\) is \(1/L\)-strongly convex, we have, for all \( A_1 \) in the domain...
of $f^*, f^*(A_1) \geq f^*(\hat{A}) + \langle \nabla f^*(\hat{A}), A_1 - \hat{A} \rangle + (1/2L)\|A_1 - \hat{A}\|^2$. This implies

$$
(\nabla f^*(\hat{A}) - Y_{k+1}^k, A_1 - \hat{A})
\leq f^*(A_1) - f^*(\hat{A}) - \langle Y_{k+1}^k, A_1 - \hat{A} \rangle - \frac{\|A_1 - \hat{A}\|^2}{2L}
\leq f^*(A_1) - f^*(\hat{A}) + \frac{\alpha\|Y_{k+1}^k\|^2}{2}
+ \frac{\|A_1 - \hat{A}\|^2}{2}\alpha - \frac{\|A_1 - \hat{A}\|^2}{2L},
$$

where the last inequality follows by using Peter-Paul inequality on $-(Y_{k+1}^k, A_1 - \hat{A})$, where $\alpha > 0$ is arbitrary. Similarly, using the fact that $f^*$ is $1/\mu$-smooth, we have, for all $A_2$ in the domain of $f^*$,

$$
(\nabla f^*(\hat{A}) - Y_{k+1}^k, \hat{A} - A_2)
\leq f^*(\hat{A}) - f^*(A_2) + \frac{\beta\|Y_{k+1}^k\|^2}{2}
+ \frac{\|A_2 - \hat{A}\|^2}{2\beta} + \frac{\|A_2 - \hat{A}\|^2}{2\mu},
$$

where $\beta > 0$ is arbitrary. Adding the two inequalities above with $A_1 = -A^*$, $A_2 = -A^*k$, we have

$$
(\nabla f^*(\hat{A}) - Y_{k+1}^k, A^{k+1} - A^*)
\leq f^*(-A^*) - f^*(-A^{k+1}) + \frac{\alpha + \beta}{2}\|Y_{k+1}^k\|^2
+ \left(\frac{1}{2\alpha} - \frac{1}{2L}\right)\|A^{k+1} + \rho\Delta Y_{k+1}^k\|^2
+ \left(\frac{1}{2\beta} + \frac{1}{2}\mu\right)\|\rho\Delta Y_{k+1}^k\|^2.
$$

Substituting this in (27), and then substituting (27) in (26), we obtain

$$
\|\hat{A}^{k+1}\|^2
\leq \|\hat{A}^k\|^2 + 2\rho(f^*(-A^*) - f^*(-A^{k+1})) + (\alpha + \beta)\|Y_{k+1}^k\|^2
+ \rho\left(\frac{1}{\alpha} - \frac{1}{L}\right)\|\hat{A}^{k+1} + \rho\Delta Y_{k+1}^k\|^2
+ \rho\left(\frac{1}{\beta} + \frac{1}{\mu}\right)\|\rho\Delta Y_{k+1}^k\|^2.
$$

Choose $\alpha = L$. Further, since $f^*$ is $1/L$-strongly convex,

$$
f^*(-A^{k+1}) \geq f^*(-A^*) + \langle \nabla f^*(-A^*), -A^{k+1} + A^* \rangle
+ \frac{\|A^{k+1} - A^*\|^2}{2L}.
$$

where $\nabla f^*(-A^*) - A^{k+1} + A^* = 0$, since $\nabla f^*(-A^*) = X^* = (1,n_1^k/n)X^*$ and $\nabla f^*(-A^*) = (1,n_1^k/n)A^{k+1} = (1,n_1^k/n)A^*$. Thus, $f^*(-A^*) - f^*(-A^{k+1}) \leq -\frac{1}{2L}\|\hat{A}^{k+1}\|^2$.

Substituting the inequality above in (28), dividing by $(\alpha + \beta)^2$ on both sides and taking supremum over $k = 0, \ldots, K - 1$, we obtain

$$
\left(1 + \frac{\rho}{L} - \frac{1}{\lambda^2}\right)(\|\hat{A}\|^{\lambda,K})^2
\leq \rho(L + \beta)(\|Y_{k+1}^k\|^{\lambda,K})^2 + \rho^3\left(\frac{1}{\beta} + \frac{1}{\mu}\right)(\|\Delta Y\|^{\lambda,K})^2 + c_3
$$

for all $K \geq 0$, for some constant $c_3 \geq 0$. From (14), we know that $1 + \rho/L - 1/\lambda^2 > 0$. Hence, using Lemma 3, we have

$$
\sqrt{1 + \frac{\rho}{L} - \frac{1}{\lambda^2}}\|\hat{A}\|^{\lambda,K}
\leq \sqrt{\rho(L + \beta)}\|Y_{k+1}^k\|^{\lambda,K} + \sqrt{\rho^3\left(\frac{1}{\beta} + \frac{1}{\mu}\right)(\|\Delta Y\|^{\lambda,K})^2 + c_3}
$$

for all $K \geq 0$. Substituting (25) in (29), we obtain that

$$
\sqrt{1 + \frac{\rho}{L} - \frac{1}{\lambda^2}}\|\hat{A}\|^{\lambda,K}
\leq \left(\sqrt{\rho(L + \beta)} + \sqrt{\rho^3\left(\frac{1}{\beta} + \frac{1}{\mu}\right)\frac{2(1 + \lambda)}{\lambda\mu - 2\rho(1 + \lambda)}}\right)\|Y_{k+1}^k\|^{\lambda,K} + c_4
$$

for all $K \geq 0$, for some $c_4 \geq 0$. Rearranging the terms, we obtain that

$$
\tilde{e}\|\hat{A}\|^{\lambda,K} \leq \left(\sqrt{\rho(L + \beta)} + \sqrt{\rho^3\left(\frac{1}{\beta} + \frac{1}{\mu}\right)\frac{2(1 + \lambda)}{\lambda\mu - 2\rho(1 + \lambda)}}\right)\|Y_{k+1}^k\|^{\lambda,K} + c_4
$$

where

$$
\tilde{e} := \sqrt{1 + \frac{\rho}{L} - \frac{1}{\lambda^2}} - \sqrt{\rho^3\left(\frac{1}{\beta} + \frac{1}{\mu}\right)\frac{2(1 + \lambda)}{\lambda\mu - 2\rho(1 + \lambda)}}.
$$

It can be verified that given (14), since

$$
\rho < \frac{\lambda^2\mu^2}{4(1 + \lambda)}\left(\frac{\lambda\mu + L(1 + \lambda)}{\lambda^2 + 1}\right),
$$

we have $\tilde{e} > 0$. Hence,

$$
\|\hat{A}\|^{\lambda,K} \leq c_5\|Y_{k+1}^k\|^{\lambda,K} + c_6
$$

(30) for all $K \geq 0$, for some $c_5 \geq 0$ and

$$
c_5 := \frac{1}{\tilde{e}}\left(\sqrt{\rho(L + \beta)} + \sqrt{\rho^3\left(\frac{1}{\beta} + \frac{1}{\mu}\right)\frac{2(1 + \lambda)}{\lambda\mu - 2\rho(1 + \lambda)}}\right) > 0.
$$

Substituting the inequality above in (25), we obtain

$$
\|\Delta Y\|^{\lambda,K} \leq \left(\frac{\mu(1 + \lambda)}{\lambda\mu - 2\rho(1 + \lambda)}\right).
$$
\[ + \frac{c_2}{\mu} \left( \frac{\mu(1+\lambda)}{\lambda+2\rho(1+\lambda)} \right) \| \bar{Y}_k \|^2 + c_7 \]

for all \( K \geq 0 \), for some \( c_7 \geq 0 \). Thus, for all \( K \geq 0 \),
\[ \| \bar{A} + \rho \Delta Y \|^2 \leq \| \bar{A} \|^2 + c_5 \| \Delta Y \|^2 + c_8 \]

for some \( c_8 \geq 0 \) and
\[ \gamma_3 := c_5 + \rho \left( \frac{\mu(1+\lambda)}{\lambda+2\rho(1+\lambda)} \right) \]
\[ = c_5 + \rho \left( 1 + \frac{2c_2}{\mu} \left( \frac{\mu(1+\lambda)}{\lambda+2\rho(1+\lambda)} \right) \right) \]
\[ = c_5 + \rho \left( \frac{\mu(1+\lambda)}{\mu \rho - \lambda \rho} \right) \]
\[ = c_5 + \rho \left( 1 + \frac{2c_2}{\mu} \left( \frac{\mu(1+\lambda)}{\mu \rho - \lambda \rho} \right) \right) \]
\[ \leq \gamma_3 \| \bar{Y}_k \|^2 + c_5 \| \Delta Y \|^2 + c_8 \]

for some \( c_8 \geq 0 \) and
\[ \gamma_3 := c_5 + \rho \left( \frac{\mu(1+\lambda)}{\lambda+2\rho(1+\lambda)} \right) \]
\[ = c_5 + \rho \left( \frac{\mu(1+\lambda)}{\lambda+2\rho(1+\lambda)} \right) \]
\[ = c_5 + \rho \left( 1 + \frac{2c_2}{\mu} \left( \frac{\mu(1+\lambda)}{\lambda+2\rho(1+\lambda)} \right) \right) \]
\[ = c_5 + \rho \left( \frac{\mu(1+\lambda)}{\mu \rho - \lambda \rho} \right) \]
\[ \leq \gamma_3 \| \bar{Y}_k \|^2 + c_5 \| \Delta Y \|^2 + c_8 \]

for some \( c_8 \geq 0 \) and
\[ \gamma_3 := c_5 + \rho \left( \frac{\mu(1+\lambda)}{\lambda+2\rho(1+\lambda)} \right) \]
\[ = c_5 + \rho \left( 1 + \frac{2c_2}{\mu} \left( \frac{\mu(1+\lambda)}{\lambda+2\rho(1+\lambda)} \right) \right) \]
\[ = c_5 + \rho \left( \frac{\mu(1+\lambda)}{\mu \rho - \lambda \rho} \right) \]
\[ \leq \gamma_3 \| \bar{Y}_k \|^2 + c_5 \| \Delta Y \|^2 + c_8 \]

for some \( c_8 \geq 0 \) and
\[ \gamma_3 := c_5 + \rho \left( \frac{\mu(1+\lambda)}{\lambda+2\rho(1+\lambda)} \right) \]
\[ = c_5 + \rho \left( 1 + \frac{2c_2}{\mu} \left( \frac{\mu(1+\lambda)}{\lambda+2\rho(1+\lambda)} \right) \right) \]
\[ = c_5 + \rho \left( \frac{\mu(1+\lambda)}{\mu \rho - \lambda \rho} \right) \]
\[ \leq \gamma_3 \| \bar{Y}_k \|^2 + c_5 \| \Delta Y \|^2 + c_8 \]

for some \( c_8 \geq 0 \) and
\[ \gamma_3 := c_5 + \rho \left( \frac{\mu(1+\lambda)}{\lambda+2\rho(1+\lambda)} \right) \]
\[ = c_5 + \rho \left( 1 + \frac{2c_2}{\mu} \left( \frac{\mu(1+\lambda)}{\lambda+2\rho(1+\lambda)} \right) \right) \]
\[ = c_5 + \rho \left( \frac{\mu(1+\lambda)}{\mu \rho - \lambda \rho} \right) \]
\[ \leq \gamma_3 \| \bar{Y}_k \|^2 + c_5 \| \Delta Y \|^2 + c_8 \]

for some \( c_8 \geq 0 \) and
\[ \gamma_3 := c_5 + \rho \left( \frac{\mu(1+\lambda)}{\lambda+2\rho(1+\lambda)} \right) \]
\[ = c_5 + \rho \left( 1 + \frac{2c_2}{\mu} \left( \frac{\mu(1+\lambda)}{\lambda+2\rho(1+\lambda)} \right) \right) \]
\[ = c_5 + \rho \left( \frac{\mu(1+\lambda)}{\mu \rho - \lambda \rho} \right) \]
\[ \leq \gamma_3 \| \bar{Y}_k \|^2 + c_5 \| \Delta Y \|^2 + c_8 \]