Asynchronous Distributed Optimization via Dual Decomposition and Block Coordinate Ascent

Yankai Lin, Iman Shames, and Dragan Nešić

***Abstract*— We study a class of distributed optimization problems of minimizing the sum of potentially non-differentiable convex objective functions (without requiring strong convexity). A novel approach to the analysis of asynchronous distributed optimization is developed. An iterative algorithm based on dual decomposition and block coordinate ascent is implemented in an edge based manner. We extend available results in the literature by allowing multiple and potentially overlapping blocks to be updated at the same time with non-uniform probabilities assigned to different blocks. Sublinear convergence with probability one is proved for the algorithm under the aforementioned weak assumptions. A numerical example is provided to illustrate the effectiveness of the algorithm.**

I. INTRODUCTION

Distributed optimization is becoming increasingly important as it has wide applications in areas such as resource allocation [22] (e.g. power networks, communication networks, robotics, and so on), distributed learning [15], distributed model predictive control (MPC) [4], and many others. We adopt the view presented in [2] where numerical computations are assigned to a group of processors (or agents) that exchange their information with others synchronously or asynchronously to minimize an objective function of interest. For a more recent survey on distributed optimization see [24].

The available results on distributed optimization can be roughly classified as primal methods and dual methods. Primal methods are largely based on consensus theory where agents solve their own local problems independently while guaranteeing that all local decision variables converge to the solution of the distributed optimization problem [13]. Dual methods rely on the idea of dual decomposition where the agents share and update common dual variables. These common dual variables under certain assumptions then will converge to the dual optimal variable which corresponds to a primal optimal solution under strong duality. Distributed dual methods are studied under different assumptions such as fixed communication network with and without delays [5], [9], [18], [19] and time-varying networks [6]. The majority of these methods assume synchronous communication and computation.

Asynchrony in communication and computation is more realistic because different agents or processors may have different computational capabilities. Moreover, their communication also depends on the bandwidth and capacity of the network which may be affected by potential packet dropouts [17]. Also, if the size of the network is large, it may be costly to synchronize the network.

In this work, we study a class of distributed dual subgradient[1] algorithms for minimizing the sum of convex objective functions that have partially overlapping dependences. This enables our algorithm to solve distributed MPC problems with semi-definite costs. Our contributions can be summarized as follows:

- We assume only convexity of the objective function without appealing to its differentiability. Consequently, we in general do not have differentiability of the dual objective function. The majority of existing literature relies on stronger assumptions such as Lipschitz or bounded gradients of the primal objective function [13], [14], [23].
- We establish that this problem can be reformulated as block coordinate decent for a potentially non-differentiable convex function. This is different from the literature including [21], [14] and [11].
- We prove almost sure convergence of the algorithm under the assumption that each block is to be activated with equal probability by allowing the overlapping blocks with non-uniform activating probability. This is different from the set-up considered in [14], where each block is activated with uniform probability. Moreover, an estimate of convergence rate is provided.

By applying dual decomposition to the problem, we formulate the dual problem which has a separable structure. This enables us to assign a primal problem that is of a smaller size to each agent. In order to perform the dual update, the pair of agents that correspond to two coupled functions need to exchange their solutions to their primal problems. We allow these communications to be asynchronous by allowing only a subset of dual variables to be updated in one iteration, thus saving communication resources. We model this asynchrony in a stochastic setting that will be detailed in Section III. We show that this set-up results in a random block coordinate ascent problem for a concave function [11], [21], and we provide sufficient conditions to show almost sure convergence to the solution of the dual problem. Under strong duality we also solve the primal problem in a stochastic setting.

This work was supported by the Australian Research Council under the Discovery Project DP170104099.

The authors are with the Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville, 3010, Victoria, Australia (email: yankail@student.unimelb.edu.au; iman.shames@unimelb.edu.au; dnesic@unimelb.edu.au).

[1]With some abuse of notations, we use ascent/descent (although our update may not necessarily be an ascent/descent direction) and subgradient/supergradient interchangeably in this work due to the relationship between convexity and concavity.

The most closely related works include [23], where an asynchronous distributed gradient method is proposed based on dynamic average consensus. Heterogeneous constant stepsizes are used for agents in the network and each agent is allowed to have its own local clock to trigger communication. Non-uniform stepsizes are also used in [8] to account for the asynchronous nature of the network. In [20], the distributed ADMM is extended to allow asynchronous updates while achieving an $O(1/k)$ convergence rate. Another related work is [7], where a fully asynchronous and distributed algorithm is proposed to solve problems in which both the objective function and the constraints may be non-convex using the method of multipliers. It is shown in the paper that the resulting distributed algorithm is equivalent to block coordinate descent for the augmented Lagrangian.

The subsequent sections are organized as follows: In Section II, we give the precise problem formulation and the corresponding dual optimization problem. In Section III, we develop the asynchronous block coordinate ascent algorithm based on the dual optimization problem whose convergence analysis is provided in Section IV. In Section V, we give a numerical example to illustrate our theoretical results. Finally, conclusions and possible future research directions are given in Section VI.

Notations: Let $\mathbb{R}$ be the set of real numbers and $\mathbb{R}^n$ be the n-dimensional Euclidean space. $||\cdot||$ denotes the Euclidean norm of a vector $x \in \mathbb{R}^n$ and the induced norm of a matrix, respectively, and $|\cdot|$ denotes the cardinality of a set. The m dimensional identity matrix is denoted by I_m and m will be omitted when it does not cause any confusion. For a convex (resp., concave) function f, ∂f denotes its subdifferential (resp., superdifferential) set.

II. PROBLEM FORMULATION

We consider an optimization problem, where N individual agents are employed to cooperatively minimize the sum of N individual objective functions that have partially overlapping dependences. The interconnection of the agents is described by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ [10], where $\mathcal{V} = \{1, 2, ..., N\}$ is the set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. An edge $(i, j) \in \mathcal{E}$ means that agent i is able to access information from agent j. The problem we study is formally given by

$$\begin{aligned} \min_{x\in\mathbb{R}^n} F(x) &= \sum_{i=1}^{N} f_i(x_i) \\ \text{s.t. } E_{ij}x_i - E_{ji}x_j &= 0,\ \forall (i,j) \in \mathcal{E}, \end{aligned} \tag{1}$$

where $x_i \in \mathbb{R}^{n_i}$ is the local variable held by agent $i \in \mathcal{V}$, $x = [x_1^T\ x_2^T\ ...\ x_N^T]^T \in \mathbb{R}^n$, $\sum_{i=1}^N n_i = n$ is the collection of all local variables, $f_i : \mathbb{R}^{n_i} \to \mathbb{R}$ is the individual objective function of agent i that satisfies the following assumption

Assumption 1: Each individual objective function f_i is convex. ■

The matrices E_{ij} and E_{ji} are selection matrices for the pair of agents i, j that characterize how the individual functions share common variables. For example, if we have $F(x) = f_1(x_1, x_2) + f_2(x_2, x_3, x_4)$ for some scalars x_1, x_2, x_3 and x_4, then we have $E_{12} = [0\ 1]$ and $E_{21} = [1\ 0\ 0]$. If we have $n_1 = n_2 = ... = n_N = \frac{n}{N}$, and $E_{ij} = I_{\frac{n}{N}}, \forall i, j \in \mathcal{V}$, the problem considered becomes the optimal consensus problem [13]. We now introduce the dual variables λ_{ij}, for agents $i, j \in \mathcal{V}$ that share common variables, which is associated with the constraint $E_{ij}x_i = E_{ji}x_j$. To prevent unnecessary communications, agents that do not share any common variables are not required to communicate. Therefore, there is no communication link between such agents, i.e. $(i, j) \notin \mathcal{E}$. Consequently, the communication graph $\mathcal{G}$ may not be complete or connected. As a result, we use an edge-based communication structure by assigning a dual vector λ_{ij} corresponding to constraint $E_{ij}x_i - E_{ji}x_j = 0$ and edge (i, j) to agent i if $(i, j) \in \mathcal{E}^+ := \{(i, j) \in \mathcal{E} : i < j\}$ and to j otherwise[2]. For notational simplicity, we introduce the vector $\lambda = [..., \lambda_{ij}^T, ...]^T, \forall (i, j) \in \mathcal{E}^+$ that contains all the dual variables. The Lagrangian L is given by $L(x, \lambda) := \sum_{i=1}^N f_i(x_i) + \sum_{(i,j)\in\mathcal{E}^+} \lambda_{ij}^T(E_{ij}x_i - E_{ji}x_j)$. The corresponding dual function is the infimum with respect to the primal variables, which may not be differentiable under Assumption 1

$$Q(\lambda) := \inf_{x\in\mathbb{R}^n} L(x, \lambda). \tag{2}$$

Due to the finite sum structure of $F(x)$, (2) can be re-written as $Q(\lambda) = \sum_{i=1}^N q_i(\lambda)$, where the individual functions q_i are given by

$$q_i(\lambda) = \inf_{x_i\in\mathbb{R}^{n_i}} f_i(x_i) + \sum_{j\in\mathcal{N}_i^+} \lambda_{ij}^T E_{ij}x_i - \sum_{j\in\mathcal{N}_i^-} \lambda_{ji}^T E_{ij}x_i, \tag{3}$$

where $\mathcal{N}_i^+ = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}^+\}$ and $\mathcal{N}_i^- = \{j \in \mathcal{V} : (j, i) \in \mathcal{E}^+\}$. It can be seen that the functions $q_i(\lambda)$ depend on the dual variable λ only, and the infimum is taken over local variables x_i. This allows the agents to individually, locally compute $q_i(\lambda)$. The Lagrange dual problem is given by

$$\max_{\lambda}\ Q(\lambda). \tag{4}$$

We further assume the following:

Assumption 2: There exists at least one finite λ^* such that $q^* := Q(\lambda^*) = \max_{\lambda} Q(\lambda)$. Moreover, for real λ there exists at least one bounded x that minimizes $L(x, \lambda)$. ■

Remark 1: The convexity assumption on individual functions f_i ensures that F is also convex. Thus, Assumption 2 ensures that there exists an optimum $x^* \in \mathbb{R}^n$ to the primal problem (1). Furthermore, problem (1) is convex with linear equality constraints and hence the Slater's conditions hold [3]. This means the primal solutions can be computed by solving (3) individually, after the dual solution is found by solving (4). ■

[2]Depending on different ways of writing the equality constraint in (1), there are multiple ways of possible dual variable assignments resulting in different communication graphs. However, the corresponding problems are mathematically equivalent to the one discussed in our paper and can be analyzed using the same approach with some notational changes only.

III. Asynchronous Block Coordinate Ascent Algorithm

In this section, we address the dual problem (4) following a novel approach where the updates of the dual variables are viewed as updates in an asynchronous block coordinate ascent algorithm. Note that, the standard supergradient update for the dual problem is as follows

$$\lambda(k+1) \in \lambda(k) + \alpha(k)\partial Q(\lambda(k)), \tag{5}$$

where $\lambda(k)$ is the value of λ at iteration k and $\alpha(k) > 0$ is the stepsize. It can be shown that (see [14] for example), the above supergradient update (5) for a single vector containing dual variables λ_{ij} is given by

$$\lambda_{ij}(k+1) = \lambda_{ij}(k) + \alpha(k)(E_{ij}x_i^*(k) - E_{ji}x_j^*(k)), \tag{6}$$

where $x_i^*(k)$ is a solution of the primal problem for agent i given $\lambda(k)$.

Definition 1: The set of edges (i,j) are called *activated* at time instance k if agent i obtains $x_j^*(k)$ from agent j via (i,j) at k and are denoted by $\mathcal{A}_k$. Otherwise, the edges are called *idle* at k and are denoted as $\mathcal{I}_k$. ■

The edge based distributed algorithm is implemented in the following way and is formally given in Algorithm 1:

i) Each agent i individually solves its primal problem (3) using dual variables available to i so far to get the solution x_i^*.
ii) Agents i and j such that $(i,j) \in \mathcal{A}_k$ exchange their local variables x_i^* and x_j^* via the edge (i,j).
iii) For such agents, an update for λ_{ij} given by (6) is performed which can be seen as an update of a block of the dual variables [21].

Algorithm 1 Asynchronous Block Coordinate Ascent Algorithm

1: **Initialization**: Assign $\lambda_{ij}(0)$ to agents i such that $(i,j) \in \mathcal{E}$.
2: **Primal Update**: For all agents $i \in \mathcal{V}$, compute

$$x_i(k) := \underset{x_i \in \mathbb{R}^{n_i}}{\arg\min}\ f_i(x_i) + \sum_{j \in \mathcal{N}_i^+} \lambda_{ij}^T(k) E_{ij} x_i - \sum_{j \in \mathcal{N}_j^-} \lambda_{ji}^T(k) E_{ij} x_i.$$

3: **Dual Update**: For all edges $(i,j) \in \mathcal{E}$ that are **activated**, i.e. $(i,j) \in \mathcal{A}_k$, the corresponding agents i and j talk to each other

$$\lambda_{ij}(k+1) = \lambda_{ij}(k) + \alpha(k)(E_{ij}x_i(k) - E_{ji}x_j(k)).$$

For all edges $(i,j) \in \mathcal{E}$ that are **idle**, i.e. $(i,j) \in \mathcal{I}_k$

$$\lambda_{ij}(k+1) = \lambda_{ij}(k).$$

4: Set $k \to k+1$ and go to Step 2.

Remark 2: We do not require that all neighboring agents exchange information at the same time. Instead, we consider an asynchronous implementation of the distributed algorithm where we activate only a subset of the edges which can be from all possible combinations of the edges. Moreover, it is assumed that the local primal problems (3) are solved by the agents in finite time with negligible errors so that activated edges perform the exact dual update (6). We refer the readers to [12] for a detailed discussion on dual decomposition using approximate primal solutions. ■

Since we have a total of $|\mathcal{E}^+|$ constraints, therefore, there are $2^{|\mathcal{E}^+|}$ possible realizations of $\mathcal{A}_k$ at any k including the cases where all or none of the edges are activated. This set-up may also be used to model potential link failures and packet dropouts typically seen in networked systems [17]. We assign to each one of these $2^{|\mathcal{E}^+|}$ possibilities a random selection matrix U_i, $1 \leq i \leq 2^{|\mathcal{E}^+|}$. Each U_i is a diagonal matrix with only 1s and 0s indicating which edges are activated in the sense of Definition 1. Additionally, we define p_i to be the probability of edges corresponding to U_i being activated and the probability distribution is assumed to be i.i.d. and satisfies the following condition:

Assumption 3: The matrix $\sum_{i=1}^{2^{|\mathcal{E}^+|}} \mathbb{I}_{p_i>0} U_i$ is full rank, where $\mathbb{I}_s$ is the characteristic function that is 1 if the statement s is true and is 0 otherwise. ■

Under Assumption 3, each edge has a strictly positive probability of being activated. Otherwise, $\sum_{i=1}^{2^{|\mathcal{E}^+|}} \mathbb{I}_{p_i>0} U_i$ will have at least one 0 diagonal element. The iteration given in Algorithm 1 can now be written as

$$\lambda(k+1) \in \lambda(k) + \alpha(k)U(k)\partial Q(\lambda(k)), \tag{7}$$

where $U(k) \in \{U_i : 1 \leq i \leq 2^{|\mathcal{E}^+|}\}$ and is selected randomly.

Remark 3: Assumption 3 is weaker than those commonly used in random block coordinated descent literature where the block is selected by uniform randomization over the blocks [21], [14]. This assumption can be satisfied by assigning heterogeneous random clocks to the agents that hold dual variables or can be regarded as a way of modelling complex behaviour including computation time of the primal problem and data dropouts. ■

Before we state the main result of this paper, we state an assumption about the concave function Q which enables the convergence analysis.

Assumption 4: There exists a scalar c, such that

$$c^2(1 + \inf_{\lambda^* \in \Lambda_{opt}} ||\lambda(k) - \lambda^*||^2) \geq ||g_Q(\lambda)||^2,\ \forall k \geq 0,$$

where $g_Q(\lambda) \in \partial Q(\lambda)$ and Λ_{opt} denotes the set of maximizers of Q. ■

Remark 4: This assumption can be satisfied by requiring the inequality to hold for all λ (however, this is not necessary) and is weaker than both the bounded gradient/subgradient assumption used in [19] and the Lipschitz continuous gradient assumption used in [14]. It allows the gradient to be unbounded when λ is far away from Λ_{opt} as well as not decreasing to 0 when λ is close to Λ_{opt}. ■

IV. Convergence Analysis

A. Almost Sure Convergence

In this part we analyse the evolution of λ. As mentioned in the previous section, we formulate the dual problem as block coordinate ascent for a potentially non-differentiable concave function. First, we define the matrix $A := \sum_{i=1}^{2^{|\mathcal{E}^+|}} p_i U_i$ which characterizes the average behaviour of the supergradient of Q over the blocks and is guaranteed to be diagonal and positive definite by Assumption 3. Then, we have the following result that will be central to our analysis.

Lemma 1: Let Assumptions 1-4 hold and define $y(k) = A^{-\frac{1}{2}}\lambda(k)$, $P(y) = Q(A^{\frac{1}{2}}y)$, $\tilde{\lambda}(k) = \lambda(k) - \lambda^*$ and $\tilde{y}(k) = y(k) - y^*$. Then given the past history represented by the σ-fields $\mathcal{F}_k := \sigma(\lambda(i), i \leq k)$, we have the following inequality for Algorithm 1

$$\mathbb{E}[||\tilde{y}(k+1)||^2|\mathcal{F}_k] \leq (1+\alpha^2(k)c^2||A||^2)||\tilde{y}(k)||^2 - 2\alpha(k)(P(y^*) - P(y(k))) + \alpha^2(k)c^2,$$

where c comes from Assumption 4. ■

Proof: If $\lambda(k)$ is given, $U(k)$ is the only source of randomness in the above iteration. The iteration (7) can be written as $\lambda(k+1) = \lambda(k) + \alpha(k)U(k)A^{-\frac{1}{2}}A^{\frac{1}{2}}g_Q(\lambda(k))$. On the other hand, we have $\partial P(y) = A^{\frac{1}{2}}\partial Q(A^{\frac{1}{2}}y) = A^{\frac{1}{2}}\partial Q(\lambda)$. This implies that for y^* such that $\partial P(y^*) = 0$, we have $A^{\frac{1}{2}}\partial Q(A^{\frac{1}{2}}y^*) = 0$. Hence $y^* = A^{-\frac{1}{2}}\lambda^*$. Consequently, (7) can be written as $\tilde{y}(k+1) = \tilde{y}(k) + \alpha(k)A^{-\frac{1}{2}}U(k)A^{-\frac{1}{2}}g_P(y(k)) = \tilde{y}(k) + \alpha(k)A^{-1}U(k)g_P(y(k))$ by left multiplying $A^{-\frac{1}{2}}$ on both sides of (7) where $g_P \in \partial P$. Using the fact that $U^2(k) = U(k)$, we have

$$\begin{aligned}
&\mathbb{E}[||\tilde{y}(k+1)||^2|\mathcal{F}_k] \\
&= \mathbb{E}[||\tilde{y}(k) + \alpha(k)A^{-1}U(k)g_P(y(k))||^2|\mathcal{F}_k] \\
&= ||\tilde{y}(k)||^2 + 2\alpha(k)[A^{-1}Ag_P(y(k))]^T\tilde{y}(k) \\
&\quad + \mathbb{E}[\alpha^2(k)[g_P(y(k))]^TA^{-2}U^2(k)g_P(y(k))|\mathcal{F}_k] \\
&= ||\tilde{y}(k)||^2 + 2\alpha(k)[g_P(y(k))]^T\tilde{y}(k) \\
&\quad + \alpha^2(k)[g_P(y(k))]^TA^{-1}g_P(y(k)).
\end{aligned}$$

Due to the fact that P is a concave function of y, we can upper bound the second term in above inequality

$$\mathbb{E}[||\tilde{y}(k+1)||^2|\mathcal{F}_k] \leq ||\tilde{y}(k)||^2 + \alpha^2(k)||g_Q(\lambda(k))||^2 - 2\alpha(k)[P(y^*) - P(y(k))]. \quad (8)$$

Assumption 4 and the fact that $g_P(y) = A^{\frac{1}{2}}g_Q(\lambda)$ yield

$$\begin{aligned}
&\mathbb{E}[||\tilde{y}(k+1)||^2|\mathcal{F}_k] \leq ||\tilde{y}(k)||^2 - 2\alpha(k)[P(y^*) - P(y(k))] \\
&\quad + \alpha^2(k)c^2||\tilde{\lambda}(k)||^2 + \alpha^2(k)c^2 \\
&\leq (1+\alpha^2(k)c^2||A||)||\tilde{y}(k)||^2 - 2\alpha(k)[P(y^*) - P(y(k))] \\
&\quad + \alpha^2(k)c^2,
\end{aligned}$$

which proves Lemma 1. ■

Next, we state the supermartingale convergence result by Robbins and Siegmund [1].

Lemma 2: Let $(\Omega, \mathcal{F}, \mathcal{P})$ be a probability space and $\mathcal{F}_0 \subset \mathcal{F}_1 \subset \cdots \subset \mathcal{F}_k$ be a sequence of σ-subfields of $\mathcal{F}$. In addition, let $v(k)$, $a(k)$, $w(k)$ and $u(k)$ be nonnegative random variables and let the following relation hold with probability one for any $k \geq 0$

$$\mathbb{E}[v(k+1)|\mathcal{F}_k] \leq (1+a(k))v(k) - u(k) + w(k), \quad (9)$$

where $\sum_{k=0}^{\infty} a(k) < \infty$ and $\sum_{k=0}^{\infty} w(k) < \infty$ are satisfied almost surely. Then the sequence $v(k)$ will converge to some random variable v almost surely and we further have $\sum_{k=0}^{\infty} u(k) < \infty$ almost surely. ■

Now we are ready to state our first main result, which states that Algorithm 1 almost surely converges to one of the optimal points.

Theorem 1: Consider the sequences $\lambda(k)$ generated by Algorithm 1. Suppose Assumptions 1–4 hold. Then, for stepsizes satisfying

$$\sum_{k=0}^{\infty} \alpha(k) = \infty \text{ and } \sum_{k=0}^{\infty} \alpha^2(k) < \infty, \quad (10)$$

we have $\lim_{k\to\infty} ||y(k) - y^*|| = 0$ for some $y^* \in Y_{opt}$ and $\lim_{k\to\infty} P(y(k)) = p^*$ almost surely, where $p^* := P(y^*) = \max_y P(y)$ and Y_{opt} denotes the set of maximizers of P. Moreover, we have $\lim_{k\to\infty} ||\lambda(k) - \lambda^*|| = 0$ for some $\lambda^* \in \Lambda_{opt}$ and $\lim_{k\to\infty} Q(\lambda(k)) = q^*$ almost surely. ■

Proof: Since we have $\sum_{k=0}^{\infty} a(k) = \infty$ and $\sum_{k=0}^{\infty} a^2(k) < \infty$, from the result of Lemma 1, we can take the sequence $||\tilde{\lambda}(k)||^2$, $2\alpha(k)(P(y^*) - P(y(k)))$ and $\alpha^2(k)c^2$ as sequences $v(k)$, $u(k)$ and $w(k)$ respectively in (9). Furthermore, $a(k)$ in (9) is taken as $a(k) = \alpha^2(k)c^2, \forall k \geq 0$. Then by Lemma 2, we have that $\sum_{k=0}^{\infty} 2\alpha(k)(P(y^*) - P(y(k))) < \infty$ and $||\tilde{y}(k)||^2$ will converge to some random variable almost surely. This implies

$$\limsup_{k\to\infty} P(y(k)) = p^*. \quad (11)$$

Moreover, we have the sequence $y(k)$ is bounded almost surely. Consider an arbitrary sample trajectory of $y(k)$. Due to the continuity of the function P, the sequence $y(k)$ must have a limit point $\bar{y}$ being an optimal point, i.e. $P(\bar{y}) = p^*$. Since the choice of y^* is arbitrary from Y_{opt}, it is without loss of generality to take $y^* = \bar{y}$ and have $||y(k) - \bar{y}|| \to 0$ almost surely. Then $y(k) \to \bar{y}$ on this sample trajectory. As a result, we have that $y(k)$ converges almost surely to a point in Y_{opt}. Because we have $y(k) = A^{-\frac{1}{2}}\lambda(k)$, $P(y) = Q(A^{\frac{1}{2}}y)$, we also have $\lim_{k\to\infty} ||\lambda(k) - \lambda^*|| = 0$ for some $\lambda^* \in \Lambda_{opt}$ and $\lim_{k\to\infty} Q(\lambda(k)) = q^*$ almost surely. ■

Remark 5: In many applications, it is desired that the agents and edges have uncoordinated stepsizes. In the case where activated edges in any iteration share the same positive end ((i,j) such that $j \in \mathcal{N}_i^+$), it is possible to guarantee the condition (10) by requiring all local stepsizes to satisfy (10). The overall stepsizes used must contain infinitely many terms of at least one local stepsize sequences. On the other hand, the summation of the squares is upper bounded by the summations of squares of all the edges. Thus (10) is ensured. This includes as a special case where only one edge

is allowed to be active per iteration which is typically used in the literature. ■

B. Convergence Rate Estimates

In this part, we move on to convergence rate analysis by stating the following Proposition.

Proposition 1: Suppose Assumptions 1-4 hold, then we have the following relationship holds almost surely:

$$\liminf_{k\to\infty} k\alpha(k)(p^* - P(y(k))) = 0. \tag{12}$$

■

Proof: The proof is done by contradiction. Assume that there exists some $\epsilon > 0$ and $\bar{k} > 0$ such that for all $k \geq \bar{k}$, we have: $k\alpha(k)(p^* - P(y(k))) \geq \epsilon$. Thus: $\alpha(k)(p^* - P(y(k))) \geq \frac{\epsilon}{k}, \forall k \geq \bar{k}$, which implies $\sum_{k=\bar{k}}^{\infty} \alpha(k)(p^* - P(y(k))) \geq \epsilon \sum_{k=\bar{k}}^{\infty} \frac{1}{k} = \infty$. This contradicts the results in Lemma 2 and Theorem 1 and the proof is complete. ■

Remark 6: The above result can be interpreted as follows. For sufficiently many iterations it is possible to observe at least one iteration where the value obtained at this iteration is arbitrarily close to q^*. However, it is not guaranteed to be true for upcoming iterations. But at least it allows us to establish some estimate of the convergence rate under Assumption 4 which is rather weak. Moreover, the convergence rate depends on how the stepsizes are chosen. Suppose we choose the stepsize such that $\alpha(k) = \frac{\delta}{(k+1)^q}$, where $\frac{1}{2} < q \leq 1$ and $\delta > 0$. Define the sequence $b(k) = \inf_{i\leq k} P(y^*) - P(y(k))$. If we require $\liminf_{k\to\infty} b(k) < \epsilon$, we have: $\liminf_{k\to\infty} k^{1-q} b(k) = 0$. Thus, we have a sublinear convergence rate of $O(\frac{1}{k^{1-q}})$ in the $\liminf$ sense. ■

It is shown in [1] that the subgradient method with fixed stepsize has a convergence rate of $O(\frac{1}{\sqrt{k}})$ to the neighborhood of the optimal value. Since we have $\frac{1}{2} < q \leq 1$, the almost sure convergence rate that can be guaranteed is no better than the subgradient method. However, this is to be expected due to the weaker assumptions we make and the fact that we are using diminishing stepsizes to ensure almost convergence to the exact optimal value. It can also be observed that. when q approaches one, the convergence rate estimates become arbitrarily slow. We now state another sublinear convergence estimate in a stochastic setting under the following assumption which can be guaranteed by requiring that $g_Q(\lambda)$ is bounded for all λ.

Assumption 5: There exists a scalar $G > 0$, such that

$$||g_Q(\lambda(k))|| \leq G,$$

for all $k \geq 0$, $g_Q \in \partial Q$. ■

The following result establishes the relationship between the number of iterations and the best estimation obtained so far.

Theorem 2: Consider the sequences $\lambda(k)$ and $y(k)$ generated by Algorithm 1. Suppose Assumptions 1-3 and Assumption 5 hold and $\sup_{y^*\in Y_{opt}} ||y(0) - y^*|| \leq R$ for some $R > 0$. Then for stepsizes such that

$$\sum_{k=0}^{\infty} \alpha(k) = \infty \text{ and } \sum_{k=0}^{\infty} \alpha^2(k) < \infty, \tag{13}$$

we have $\lim_{k\to\infty} ||y(k) - y^*|| = 0$ for some $y^* \in Y_{opt}$ and $\lim_{k\to\infty} P(y(k)) = p^*$ almost surely. Moreover, define $P_{\text{best}}(k) = \max_{i=0,1,\ldots,k} \mathbb{E}[P(y(i))|\mathcal{F}_{i-1}]$, we have $P_{\text{best}}(k)$ satisfies

$$p^* - P_{\text{best}}(k) \leq \frac{R^2 + G^2 \sum_{i=1}^{k} \alpha^2(i)}{2\sum_{i=1}^{k} \alpha(i)}$$

and thus converges to p^* almost surely. ■

Proof: Because of the way we define P and y, Assumption 5 implies that there exists another constant $G = ||A^{\frac{1}{2}}||c$ such that $||g_P(y(k))|| \leq G$. Applying the law of total expectations to (8) iteratively [2, Proposition D.5 (b), Appendix D], we have $\mathbb{E}[||\tilde{y}(k+1)||^2|\mathcal{F}_k] \leq \mathbb{E}[||\tilde{y}(k)||^2|\mathcal{F}_{k-1}] + \alpha^2(k)G^2 - 2\alpha(k)(p^* - \mathbb{E}[P(y(k))|\mathcal{F}_{k-1}]) \leq ||\tilde{y}(0)||^2 + \sum_{i=1}^{k} \alpha^2(i)G^2 - 2\sum_{i=1}^{k} \alpha(i)(q^* - \mathbb{E}[P(y(i))|\mathcal{F}_{i-1}])$,with probability 1. Since we have $\sup_{y^*\in Y_{opt}} ||y(0) - y^*|| \leq R$, it follows that: $0 \leq R^2 - 2\sum_{i=1}^{k} \alpha(i)(p^* - \mathbb{E}[P(y(i))|\mathcal{F}_{i-1}]) + \sum_{i=1}^{k} \alpha^2(i)G^2$. Then

$$p^* - P_{\text{best}}(k) \leq \frac{R^2 + G^2 \sum_{i=1}^{k} \alpha^2(i)}{2\sum_{i=1}^{k} \alpha(i)}$$

holds almost surely. Because of the assumptions on the stepsizes, $\sum_{k=0}^{\infty} a(k) = \infty$ and $\sum_{k=0}^{\infty} a^2(k) < \infty$, we establish almost sure convergence of $P_{\text{best}}(k)$ to p^*. ■

Remark 7: Based on Theorem 2, we can state an improved convergence rate for the widely used stepsizes $\alpha(k) = \frac{\delta}{k+1}$ ($\delta > 0$) compared to the one established in Proposition 1. By using the fact that $\sum_{t=0}^{k} \frac{1}{t+1} \geq \int_0^k \frac{1}{1+t} dt = \ln(t+1)|_0^k = \ln(k+1)$, a convergence rate (in a different setting) of $O(\frac{1}{\ln k})$ for this choice of stepsize is established. It is also possible to state convergence rate when the gradient ∇Q is Lipschitz continuous which can be guaranteed by assuming that the primal objective function is strongly convex. In this case, sufficiently small constant stepsizes can be used to accelerate the convergence of the algorithm and we refer the readers to [14] for related results. ■

V. A Numerical Example

In this section, we study a numerical example to illustrate our main result. In particular we consider the following optimization problem:

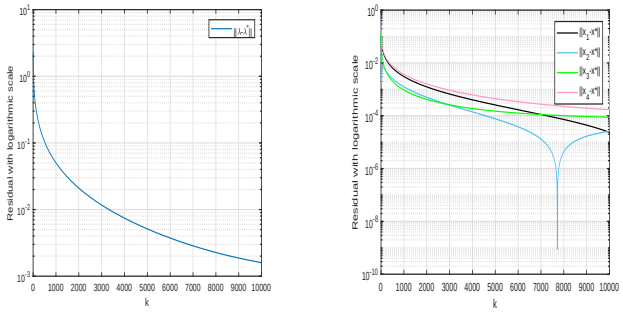
$$\min_{x\in\mathbb{R}^4} \sum_{i=1}^{4} x_i \ln 2^{i-1} x_i$$
$$\text{s.t. } x_1 - x_2 = 0, x_2 - x_3 = 0, x_3 - x_4 = 0,$$

The Lagrangian is given by

$$L(x,\lambda) = x_1 \ln x_1 + x_2 \ln 2x_2 + x_3 \ln 4x_3 + x_4 \ln 8x_4 + \lambda_{12}(x_1 - x_2) + \lambda_{23}(x_2 - x_3) + \lambda_{34}(x_3 - x_4),$$

where $x = [x_1\ x_2\ x_3\ x_4]^T$ and $\lambda = [\lambda_{12}\ \lambda_{23}\ \lambda_{34}]^T$. The ideal synchronized update for the dual variable λ is given by

$$\lambda(k+1) = \lambda(k) + \alpha(k) \begin{bmatrix} x_1^*(k) - x_2^*(k) \\ x_2^*(k) - x_3^*(k) \\ x_3^*(k) - x_4^*(k) \end{bmatrix},$$



(a) (b)

Fig. 1. (a) Plot of the residual of the dual variable $r_d = ||\lambda - \lambda^*||$ versus the number of iterations. (b) Plot of the residual of the primal variable $r_{pi} = ||x_i - x^*||$ versus the number of iterations

where $x_i^*(k)$ is a solution for the corresponding primal problem given $\lambda(k)$. Each communication of two agents via an edge is equivalent to an update on the corresponding block of the dual variable. In this example, we assume the 3 edges $(1,2)$, $(2,3)$ and $(3,4)$ are activated with the following probabilities:

- $(1,2)$ and $(2,3)$ activated: 50%;
- $(2,3)$ and $(3,4)$ activated: 30%;
- $(1,2)$ and $(3,4)$ activated: 20%;
- Other cases: 0.

The stepsize for the dual update is chosen as $\alpha(k) = \frac{4}{(k+1)^{\frac{3}{4}}}$. The simulation results are averaged over 1000 runs with each run consisting of 10000 iterations. The plots for the residuals of both dual variables and primal variables are given in Fig. 1(a) and Fig. 1(b) respectively. It can be indeed observed that, both dual and primal variables tend to λ^* and x^* as the number of iterations tends to infinity which is guaranteed by our main result with probability one.

VI. Conclusion and Future Work

We have proposed an edge based asynchronous distributed dual algorithm to minimize the sum of convex objective functions. The analysis was done by viewing the asynchronous algorithm as block coordinate ascent for a concave function. Under relatively weak assumptions on the objective function, sufficient conditions on the communications between agents and stepsizes were provided to prove almost sure convergence of the algorithm. Sublinear convergence rate estimates were stated if $\frac{1}{k+1}$-type stepsizes are used. A numerical example was given to illustrate our main result. Future work will focus on conditions that enable the use of uncoordinated stepsizes for all agents for the general case and detailed modelling of the primal updates. We are also interested in investigating deterministic protocols to activate edges such as round-robin and persistently exciting protocols.

References

[1] D. P. Bertsekas, *Convex Optimization Algorithms*. Belmont, MA, USA: Athena Scientific, 2015.

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

[3] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge Univ. Press, 2004.

[4] C. Conte, C. N. Jones, M. Morari, and M. N. Zeilinger, Distributed synthesis and stability of cooperative distributed model predictive control for linear systems, *Automatica*, vol. 69, pp. 117-125, 2016.

[5] J. C. Duchi, A. Agarwal, and M. J. Wainwright, Dual averaging for distributed optimization: Convergence analysis and network scaling, *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 592-606, 2012.

[6] A. Falsone, K. Margellos, S. Garatti, and M. Prandini, Dual decomposition for multi-agent distributed optimization with coupling constraints, *Automatica*, vol. 84, pp. 149-158, 2017.

[7] F. Farina, A. Garulli, A. Giannitrapani, and G. Notarstefano, A distributed asynchronous method of multipliers for constrained nonconvex optimization, *Automatica*, vol. 103, pp. 243-253, 2019.

[8] P. Lin, W. Ren, C. Yang, and W. Gui, Distributed optimization with nonconvex velocity constraints, nonuniform position constraints, and nonuniform stepsizes, *IEEE Transactions on Automatic Control*, vol. 64, no. 6, pp. 2575-2582, 2019.

[9] A. Makhdoumi and A. Ozdaglar, Convergence rate of distributed ADMM over networks, *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 5082-5095, 2017.

[10] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton, NJ: Princeton Univ. Press, 2010.

[11] I. Necoara, Random coordinate descent algorithms for multi-agent convex optimization over networks, *IEEE Transactions on Automatic Control*, vol. 58, no. 8, pp. 2001-2012, 2013.

[12] I. Necoara and V. Nedelcu, Rate analysis of inexact dual first-order methods application to dual decomposition, *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1232-1243, 2014.

[13] A. Nedić and A. Ozdaglar, Distributed subgradient methods for multi-agent optimization, *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48-61, 2009.

[14] I. Notarnicola, R. Carli, and G. Notarstefano, Distributed partitioned big-data optimization via asynchronous dual decomposition, *IEEE Transactions on Control of Network Systems*, vol. 5, no. 4, pp. 1910-1919, 2018.

[15] J. Predd, S. Kulkarni, and H.V. Poor, A collaborative training algorithm for distributed learning, *IEEE Transactions on Information Theory*, vol. 55, no. 4, pp. 1856-1871, 2009.

[16] P. Richtárik and M. Takáč, Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function, *Mathematical Programming*, vol. 144, no. 1/2, pp. 1-38, 2014.

[17] M. Tabbara and D. Nešić, Input-output stability of networked control systems with stochastic protocols and channels, *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1160-1175, 2008.

[18] A. Teixeira, E. Ghadimi, I. Shames, H. Sandberg, and M. Johansson, The ADMM algorithm for distributed quadratic problems: Parameter selection and constraint preconditioning, *IEEE Transactions on Signal Processing*, vol. 64, no. 2, pp. 290-305, 2016.

[19] H. Terelius, U. Topcu, and R. M. Murray, Decentralized multi-agent optimization via dual decomposition, in *Proceedings of the 18th World Congress of the International Federation of Automatic Control*, pp. 11245-11251, 2011.

[20] E. Wei and A. Ozdaglar, On the $O(1/k)$ convergence of asynchronous distributed alternating direction method of multipliers, in *Proceedings of the IEEE Global Conference on Signal and Information Processing*, pp. 551-554, 2013.

[21] S. J. Wright, Coordinate descent algorithms, *Mathematical Programming*, vol. 151, no. 1, pp. 3-34, 2015.

[22] L. Xiao, M. Johansson, and S. Boyd, Simultaneous routing and resource allocation via dual decomposition, *IEEE Transactions on Communications*, vol. 52, no. 7, pp. 1136-1144, 2004.

[23] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, Convergence of asynchronous distributed gradient methods over stochastic networks, *IEEE Transactions on Automatic Control*, vol. 63, no. 2, pp. 434-448, 2018.

[24] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, A survey of distributed optimization, *Annual Reviews in Control*, vol. 47, pp. 278-305, 2019.