ASYNCHRONOUS ONLINE ADMM FOR CONSENSUS PROBLEMS

Javier Matamoros

Centre Tecnològic de Telecomunicacions de Catalunya (CTTC / CERCA)
email: javier.matamoros@cttc.es

ABSTRACT

In this paper, we consider the consensus problem where a set of nodes cooperate to minimize a global cost. In particular, we consider an online setting and propose an online algorithm based on the alternating direction method of multipliers. Besides, we take into account the asynchronous operation of the nodes. In this context, we prove that the algorithm attains sublinear regret on the objective. Finally, we assess numerically the performance of the algorithm in a distributed sparse regression problem.

***Index Terms*—** Distributed consensus, ADMM, online learning.

1. INTRODUCTION

In recent years, we are experiencing the advent of the big data era. Big data refers to the collection, storage and processing of large volumes of data. These massive amounts of data may be either collected centrally or distributedly by a set of nodes (e.g., wireless sensor networks). Another important feature in big data is the velocity at which data is processed and interpreted. In this regard, the research community distinguishes between batch and online methods. The former refers to those methods that process large chunks of data in an offline fashion and that usually exhibit a high computational complexity. In contrast, the latter refers to lighter methods that do the processing at the same time the data is collected. This paper focuses on distributed and online learning methods where a set of nodes cooperate to learn information of the collected data in an online fashion.

Distributed optimization and learning theory have recently attracted lots of attention. In the batch setting, distributed optimization theory has been applied to fields such as wireless communications, signal processing, machine learning and smart grid (see e.g, [1] for a comprehensive survey on distributed optimization). The common ground of these works is a system composed of a set of nodes that cooperate in order to achieve a global goal. In this context, the alternating direction method of multipliers (ADMM) have became a popular technique due to its superior performance with respect to traditional primal-dual decomposition methods [2]. Distributed algorithms based on the ADMM have been proposed in [3, 4, 5]. To remark the recent works in [6, 7], where the authors propose a distributed asynchronous ADMM algorithm for (convex and non convex) consensus problems. Interestingly, the algorithm is shown to converge to the set of stationary solutions. On the other, online learning and convex optimization theory has experienced substantial advances (see [8] and references therein). In this regard, Wang and Barnejee introduced in [9] the online version of ADMM. This work analyzes the ADMM from an online perspective and shows that it attains a sublinear regret.

In this paper, we consider the distributed consensus problem where a set of nodes want to reach consensus on their actions aided by a master node. The proposed algorithm is suitable for a number of applications such as sparse distributed regression, distributed support vector machine and distributed estimation. Differently from [6], we focus on the online scenario and thus study the regret of the algorithm. Unlike [9], we further consider asynchronous nodes. Namely, nodes may be active or idle for some periods without reporting updates to the master node. This allows us to account for a number of realistic situations: communication errors, collisions in the MAC channel and nodes powered by energy harvesting. Under some mild assumptions, we prove that the algorithm attains a sublinear regret on the objective. Finally, we particularize the algorithm for the problem of sparse distributed regression with asynchronous nodes.

2. SYSTEM MODEL

Consider a communication network composed of N nodes indexed by set $\mathcal{N}$ and one master node that coordinates the network. Each node has associated a time-varying convex cost function $f_{i,t} : \mathbb{R}^m \to \mathbb{R}$, with subscripts i and t standing for the node and time slot indices, respectively. The action played by node i is denoted by vector x_i of length m. In addition, $x_i \in \mathcal{X}$ where $\mathcal{X}$ is a compact set. In this paper, we consider the *consensus* problem over a time horizon T, where nodes cooperate through the master node to reach consensus on their actions $\{x_i\}$ while minimizing the total cost. Mathematically speaking, the problem is posed as follows:

$$\min_{\{x_i \in \mathcal{X}\}, z} \quad \sum_{t=1}^{T} \left(\sum_{i=1}^{N} f_{i,t}(x_i) + h(z) \right) \tag{1a}$$

$$\text{s.t.} \quad x_i = z \quad \forall i \tag{1b}$$

with $z \in \mathbb{R}^m$ standing for the consensus variable and $h(z)$ standing for a convex function (e.g., regularization function such as ℓ_1 norm). It is worth noting that the formulation in (1) is quite general and allows us to pose a number of interesting problems such as distributed support vector machine and distributed regression (see [2] for further details).

2.1. Batch ADMM: Review

Solving (1) entails the knowledge of all cost functions $\{f_{i,t}\}_{t=1}^{T}$ from the onset. In this case, one can thus resort to a number of well-known methods to solve (1), which are referred to as batch methods by the research community. In this paper, we focus our attention on the ADMM [2]. To introduce the batch version of ADMM, let us first write the augmented Lagrangian of (1):

$$\mathcal{L}_\rho(\{x_i\}, z, \{\lambda_i\}) = \sum_{t=1}^{T} \left(\sum_{i=1}^{N} f_{i,t}(x_i) + h(z) \right) + \frac{\rho}{2} \sum_{i=1}^{N} \|x_i - z\|_2^2 + \sum_{i=1}^{N} \lambda_i^T (x_i - z) \tag{2}$$

This work is partly supported by the Spanish and Catalan Governments by grants INTENSYV (TEC2013-44591-P) and 2014-SGR-1567.

with $\{\lambda_i\}$ standing for the dual variables associated to (1b) and ρ being a positive constant. The batch ADMM alternates the following primal and dual updates until convergence

$$\{x_{i,k+1}\} = \underset{\{x_i \in \mathcal{X}\}}{\operatorname{argmin}} \mathcal{L}_\rho(\{x_i\}, z_k, \{\lambda_{i,k}\}) \tag{3}$$

$$z_{k+1} = \underset{z}{\operatorname{argmin}} \mathcal{L}_\rho(\{x_{i,k+1}\}, z, \{\lambda_{i,k}\}) \tag{4}$$

$$\lambda_{i,k+1} = \lambda_{i,k} + \rho(x_{i,k+1} - z_{k+1}) \quad \text{for} \quad i = 1, \ldots, N \tag{5}$$

where k refers to the iteration index.

The batch ADMM is suitable for decentralized implementation as discussed in the following. Nodes compute $\{x_i\}$ in parallel by exploiting the separable structure of the augmented Lagrangian. Then, the master node, upon collecting the updated $\{x_i\}$ from the nodes, computes and broadcasts the consensus variable z_{k+1} to the nodes. The procedure is repeated until convergence.

2.2. Online operation

In online optimization problems, cost functions are not available from the onset and nodes must play their actions in an online fashion. To be more precise, at time t nature reveals cost function $f_{i,t}$ to node i which uses[1] it to compute the next action, i.e,. $x_{i,t+1} \in \mathbb{R}^m$. The performance of online algorithms is assessed by the so-called regret that we define next.

Definition 1. *Let $\{\{x_{i,t+1}\}_{i=1}^N, z_t\}_{t=1}^T$ denote a sequence of actions generated by an online algorithm in a time horizon T, and let $\{x_i^\star\}, z^\star$ be the optimal batch solution of* (1). *Then, the regret of the online algorithm is defined as*

$$R = \sum_{t=1}^{T} \left(\sum_{i=1}^{N} f_{i,t}(x_{i,t}) - f_{i,t}(x_i^\star) + h(z_t) - h(z^\star) \right) \tag{6}$$

In other words, the regret measures the loss incurred by the online algorithm with respect to the optimal (batch) solution. Typically, one is interested in online algorithms that attain a sublinear regret.

2.3. Asynchronous operation

Besides the online operation, wireless communication networks impose additional constraints. To account for that, we consider an asynchronous system where only a subset of nodes $\mathcal{A}_t \subseteq \mathcal{N}$ is active at time slot t and, thus, reporting their updates to the central coordinator. Likewise, the master node only broadcasts updates of the consensus variable to the set of active nodes $\mathcal{A}_t$. This means that idle sensors, indexed by set $\mathcal{A}_t^c = \mathcal{N} \setminus \mathcal{A}_t$, do not report their updates to the master node, neither receive consensus updates from the master node[2]. This simple model allows us to account for a variety of situations: communication link outages, asynchronous operation of nodes powered by energy harvesting, computation delays due to heterogeneous nodes, and collisions when seizing the channel.

3. ASYNCHRONOUS ONLINE ADMM

Before presenting the algorithm, we introduce some assumptions needed to derive the results presented in this paper.

Assumption 1.

(a) Functions $\{f_{i,t}(x)\}$ are convex with bounded subgradients, that is, $\|\partial f_{i,t}(x)\|_2 \leq L_x$ for $i \in \mathcal{N}$.

(b) Function $h(x)$ is a convex function. Besides, $h(0) = 0$ and $h(x) \geq 0$.

(c) Let $\bar{t}_i$ be the last iteration before t where node i reported its update to the master node. Hence, the maximum number of iterations that a node is idle is upper bounded by τ, i.e., $t - \bar{t}_i + 1 \leq \tau$.

(d) Let $\{x_i^\star\}$ and $z^\star$ be the optimal batch solution, i.e, satisfying $x_i^\star = z^\star$ for $i \in \mathcal{N}$, then $\|x_i^\star\|_2^2 \leq D$ for $i \in \mathcal{N}$ and $\|z^\star\|_2^2 \leq D$. Besides, $z_0 = 0$, and $\lambda_{i,0} = 0$ and $x_{i,0} = 0$ for $i \in \mathcal{N}$.

(e) $f_{i,t}(x_i) - f_{i,t}(x_i^\star) \leq F$ for $i \in \mathcal{N}$ and a positive constant F.

In Assumption 1, *(a)* allows non-differentiable functions with bounded subgradients (e.g., ℓ_1 norm). Note that if $\mathrm{dom} f_{i,t}$ is compact this is satisfied for a large family of convex functions (e.g., ℓ_2 norm). *(b)* allows us to have regularization functions such as ℓ_1 norm. (c) prevents the sensors from having too outdated information and is needed to establish a sublinear regret bound. Assumptions *(d)* and *(e)* are quite standard in online convex and learning theory [9, 10]. Actually, Assumption (e) is satisfied for Lipschitz continuous functions [10, Section 2].

Bearing all the above in mind, the proposed algorithm produces the following updates at time slot t

- The set of active nodes , i.e., $\mathcal{A}_t$, update their local variables as follows

$$x_{i,t+1} = \underset{x \in \mathcal{X}}{\operatorname{argmin}} \sum_{l=\bar{t}_i+1}^{t} f_{l,i}(x) + x^T \lambda_{i,t} + \frac{\rho}{2}\|x - z_{\bar{t}_i}\|_2^2 + \frac{\eta}{2}\|x - x_{i,t}\|_2^2 \tag{7}$$

whereas for the set of idle nodes, i.e., $\mathcal{A}_t^c$, let $x_{i,t+1} = x_{i,t}$. Active nodes transmit $\{x_{i,t+1}\}$ to the master node.

- The master node updates its variable z as follows

$$z_{t+1} = \underset{z}{\operatorname{argmin}}\, h(z) - z^T \sum_{i \in \mathcal{A}_t} \lambda_{i,t} + \frac{\rho}{2} \sum_{i \in \mathcal{A}_t} \|x_{i,t+1} - z\|_2^2 \tag{8}$$

and broadcasts it to the set of active nodes, i.e, $\mathcal{A}_t$.

- All nodes update their dual variables

$$\lambda_{i,t+1} = \begin{cases} \lambda_{i,t} + \rho(x_{i,t+1} - z_{t+1}) & \text{for} \quad i \in \mathcal{A}_t \\ \lambda_{i,t} & \text{for} \quad i \in \mathcal{A}_t^c \end{cases} \tag{9}$$

It is worth noting that, in contrast to the batch ADMM presented in Section 2.1, nodes operate in an online fashion by playing an action at each timeslot using past information only.

3.1. Regret Analysis

The following theorem establishes the sublinear regret of the asynchronous online ADMM presented in the previous section.

[1] It may also use all the past information, i.e, $\{f_{i,l}\}_{l=1}^{t}$.

[2] Note that this is particularly interesting for energy harvesting sensors that remain idle until sufficient energy has been harvested for transmission

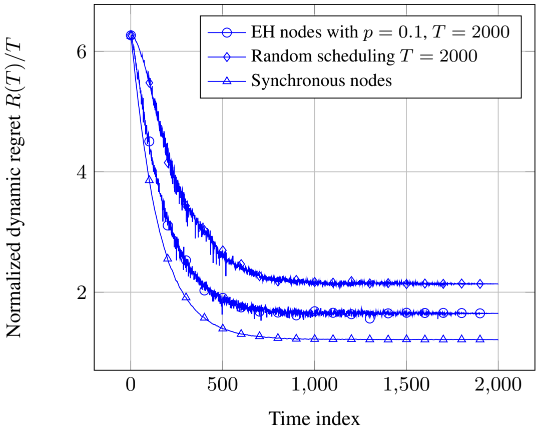

Fig. 1. Average regret vs time index for different values of the node activity probability p and parameter η. System parameters: $m = 100, N = 100, k = 5, \theta = 0.1, \rho = 0.1, \eta = 2, \sigma_w^2 = 0.1$

Theorem 1. *The sequence of iterates* $\{x_{i,t}, z_t\}$ *generated by* (7) *and* (8) *have the following sublinear regret bound on the objective:*

$$\sum_{t=0}^{T-1}\left(\sum_{i\in\mathcal{N}}(f_{i,t}(x_{i,t}) - f_{i,t}(x_i^\star)) + h(z_t) - h(z^\star)\right) \leq \frac{N\tau^2 L_x^2 + ND}{2}\sqrt{T} + \frac{ND\rho + 2N\tau F}{2} \qquad (10)$$

where $\{x_i^\star\}, z^\star$ *stand for the optimal batch solution of* (1)

Proof. See Appendix A □

4. RESULTS AND DISCUSSION

In this section, we apply the proposed algorithm to the decentralized online regression problem. In this case, variables $\{x_i\}, z$ are assumed to be sparse with only k elements different from zero. Nodes take observations of the form:

$$y_{i,t} = a_{i,t}^T x + w_{i,t} \quad t = 1, \ldots, T \quad \text{and} \quad i \in \mathcal{N} \qquad (11)$$

with $a_{i,t} \in \mathbb{R}^m$ standing for the measurement vector and $\{w_{i,t}\}$ stand for i.i.d zero mean Gaussian noise variables with variance σ_w^2. Hence, the goal is to reach consensus on a sparse vector x given the measurements in (11). To that end, we formulate the problem as a LASSO problem where functions in (1) are particularized to $f_{i,t}(x) = \frac{1}{2}\|y_{i,t} - a_{i,t}^T x\|_2^2$ for $i \in \mathcal{N}$, and $h(z) = \theta\|z\|_1$, with θ standing for a positive constant.

To assess the impact of the asynchronous operation of the nodes we consider two different scenarios: i) nodes powered by energy harvesting (EH) and ii) random scheduling of nodes. In the former scenario, nodes can only provide an update to the master when there is enough energy in its battery. For simplicity, we model the resulting node activity as an i.i.d on-off process at each node with activity probability p. As for the latter, we assume that the master node selects randomly one node per time slot to report its update. Finally, as a benchmark we also consider the synchronous case, where all sensors report (and receive) updates at each time slot.

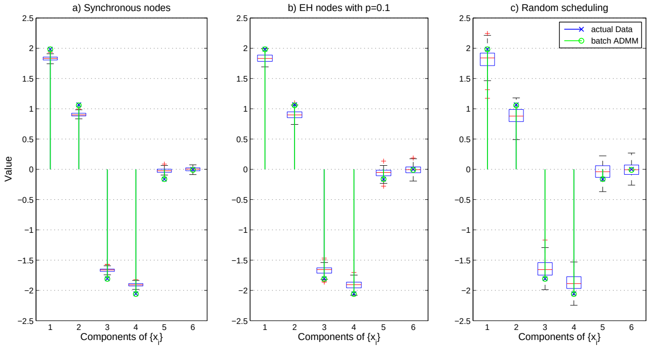

Fig. 2. Values of the k nonzero components of $\{x_i\}$ plus one of the zero components. The central mark on each box represents the median, the edges of the box represent the 25th and 75th percentiles. The whiskers extend to the rest of points and outliers are represented by red marker +. Stem plots represent the batch solution and the actual signal generated in the simulations. System parameters: $m = 100, N = 100, k = 5, \theta = 0.1, \rho = 0.1, \eta = 2, \sigma_w^2 = 0.1, T = 2000$.

In Figure 1 we plot the evolution of the (normalized) dynamic regret. The dynamic regret at time t is defined as follows:

$$R(t) = \sum_{l=1}^{t}\left(\sum_{i=1}^{N} f_{i,l}(x_{i,l}) - f_{i,l}(x_i^\star) + h(z_l) - h(z^\star)\right) + \sum_{r=t+1}^{T}\left(\sum_{i=1}^{N} f_{i,r}(x_{i,t}) - f_{i,r}(x_i^\star) + h(z_t) - h(z^\star)\right)$$

with $R(T)$ being the classical definition of regret provided in Definition 1. Essentially, at time slot t the equation above provides the regret obtained if nodes would select $x_{i,t'} = x_{i,t}$ and $z_t = z_{t'}$ for $t' > t$.

From Fig. 1, It is worth noting that all curves have the same decreasing trend. Indeed, the results show that in all cases the algorithm converges quickly to a stationary solution, since there are no major regret deviations beyond 500 time slots. As expected, the asynchronous operation of the nodes has an impact on the attained regret. This is specially pronounced for the extreme asynchronous case of random scheduling. This trend is illustrated in Figure 2 which depicts the deviations in the values of $\{x_i\}$ computed at the nodes after T time slots. Clearly, in the case of synchronous nodes, the values of the components of $\{x_i\}$ are well concentrated around the median whereas larger deviations occurs for the asynchronous scenarios.

In conclusions, in this paper we have proposed and analyzed an asynchronous online optimization algorithm for consensus problems. The proposed method is based on the ADMM and is shown to attain a sublinear regret with respect to its batch counterpart. The algorithm has been applied to the decentralized online regression problem with asynchronous nodes. Numerical results reveal that the algorithm converges fast to a stationary solution. Additionally, numerical results confirm that an asynchronous operation of the nodes impacts on the attained regret of the algorithm. Future work encompasses the regret analysis of the consensus constraints.

A. PROOF OF THEOREM 1

To ease the notation, we first define $g_{i,t}(x) = \sum_{l=\bar{t}_i+1}^{t} f_{l,i}(x)$. Now, we introduce the following lemma

Lemma 1. *Under Assumption 1, at time instant t we have the following inequality*

$$\begin{aligned}\sum_{i\in\mathcal{A}_t} g_{i,t}(x_{i,t}) - g_{i,t}(x^\star) + h(z_{t+1}) - h(z^\star)\\ \leq \frac{1}{2\rho}\sum_{i\in\mathcal{A}_t}\left(\|\lambda_{i,t}\|_2^2 - \|\lambda_{i,t+1}\|_2^2\right)\\ + \frac{\rho}{2}\sum_{i\in\mathcal{A}_t}\left(\|z^\star - z_{\bar{t}_i}\|_2^2 - \|z^\star - z_{t+1}\|_2^2\right)\\ + \frac{\eta}{2}\sum_{i\in\mathcal{A}_t}\left(\|x_i^\star - x_{i,t}\|_2^2 - \|x_i^\star - x_{i,t+1}\|_2^2\right)\\ + |\mathcal{A}_t|\frac{\tau^2 L_x^2}{2\eta} \qquad (12)\end{aligned}$$

Proof. The proof follows from standard convex arguments. Derivations are omitted due to space limitation. □

From Lemma 1, we notice that taking the sum over the time horizon T, the first three terms on the RHS become telescopic sums. The next Lemma provides upper bounds for these terms and substitutes the dependence of the LHS on z^{t+1} by z^t.

Lemma 2. *Under Assumption 1 the iterates generated by the asynchronous online ADMM satisfy*

$$\begin{aligned}\sum_{t=0}^{T-1}\left(\sum_{i\in\mathcal{A}_t}\left(g_{i,t}(x_{i,t}) - g_{i,t}(x^\star)\right) + h(z_t) - h(z^\star)\right)\\ \leq \frac{\tau^2 L_x^2}{2\eta}\sum_{t=0}^{T-1}|\mathcal{A}_t| + \frac{\eta N D}{2} + \frac{N\rho D}{2} \qquad (13)\end{aligned}$$

Proof. See Appendix A.1. □

Now, for each $i \notin \mathcal{A}_{T-1}$, let $\tilde{t}_i$ be the last iteration where node i was active, then we have that

$$\begin{aligned}\sum_{t=0}^{T-1}\sum_{i\in\mathcal{A}_t}\left(g_{i,t}(x_{i,t}) - g_{i,t}(x^\star)\right)\\ = \sum_{t=0}^{T-1}\sum_{i\in\mathcal{A}_t}\sum_{l=\bar{t}_i+1}^{t}\left(f_{i,l}(x_{i,t}) - f_{i,l}(x_i^\star)\right)\\ = \sum_{t=0}^{T-1}\sum_{i\in\mathcal{N}}\left(f_{i,t}(x_{i,t}) - f_{i,t}(x_i^\star)\right)\\ - \sum_{i\notin\mathcal{A}_{T-1}}\sum_{l=\tilde{t}_i+1}^{T-1}\left(f_{i,l}(x_{i,t}) - f_{i,l}(x_i^\star)\right) \qquad (14)\end{aligned}$$

Noting that $x_{i,l} = x_{i,t}$ for $l = \bar{t}_i + 1, \ldots, t$ and using Assumption 1 *(e)*, that is $f_{i,t}(x_i) - f_{i,t}(x_i^\star) \leq F$, the last term in (14) can be bounded as follows:

$$\sum_{i\notin\mathcal{A}_{T-1}}\sum_{l=\tilde{t}_i+1}^{T-1}\left(f_{i,l}(x_{i,t}) - f_{i,l}(x_i^\star)\right) \leq N\tau F \qquad (15)$$

where we have used $|\mathcal{A}_{T-1}^c| \leq N$ and that $T - \tilde{t}_i - 1 \leq \tau$. By using (15) and (14) in (13) and noting that $|\mathcal{A}_t| \leq N\ \forall t$, we obtain:

$$\begin{aligned}\sum_{t=0}^{T-1}\left(\sum_{i\in\mathcal{N}}\left(f_{i,t}(x_{i,t}) - f_{i,t}(x_i^\star)\right) + h(z_t) - h(z^\star)\right)\\ \leq \frac{N\tau^2 L_x^2 T}{2\eta} + \frac{N\eta D}{2} + \frac{N\rho D}{2} + N\tau F \qquad (16)\end{aligned}$$

Finally, by letting $\eta = \sqrt{T}$ concludes the proof.

A.1. Proof of Lemma 2

Consider the following telescopic sum

$$\begin{aligned}\frac{1}{2\rho}\sum_{t=0}^{T-1}\sum_{i\in\mathcal{A}_t}\left(\|\lambda_{i,t}\|_2^2 - \|\lambda_{i,t+1}\|_2^2\right)\\ = \frac{1}{2\rho}\sum_{t=0}^{T-1}\sum_{i\in\mathcal{N}}\left(\|\lambda_{i,t}\|_2^2 - \|\lambda_{i,t+1}\|_2^2\right)\\ = \frac{1}{2\rho}\sum_{i\in N}\left(\|\lambda_{i,0}\|_2^2 - \|\lambda_{i,T}\|_2^2\right)\\ \leq 0 \qquad (17)\end{aligned}$$

where, in the first step, we have used the fact that for $i \notin \mathcal{A}_t$ $\lambda_{i,t+1} = \lambda_{i,t}$ and, the last step follows from Assumption 1 *(d)* that $\lambda_{i,0} = 0$. Similarly, we have that

$$\begin{aligned}\frac{\eta}{2}\sum_{t=1}^{T-1}\sum_{i\in\mathcal{A}_t}\left(\|x_i^\star - x_{i,t}\|_2^2 - \|x_i^\star - x_{i,t+1}\|_2^2\right)\\ = \frac{\eta}{2}\sum_{t=1}^{T-1}\sum_{i\in\mathcal{N}}\left(\|x_i^\star - x_{i,t}\|_2^2 - \|x_i^\star - x_{i,t+1}\|_2^2\right)\\ \leq \frac{\eta}{2}\sum_{i\in\mathcal{N}}\|x_i^\star - x_{i,0}\|_2^2\\ \leq \frac{\eta N D}{2} \qquad (18)\end{aligned}$$

where again we have used $x_{i,t+1} = x_{i,t}$ for $i \notin \mathcal{A}_t$ and the last step follows from Assumption 1 *(d)*. Additionally, for each $i \notin \mathcal{A}_{T-1}$ define $\tilde{t}_i$ as the last iteration where node i was active, then we have that

$$\begin{aligned}\frac{\rho}{2}\sum_{t=0}^{T-1}\sum_{i\in\mathcal{A}_t}\left(\|z^\star - z_{\bar{t}_i}\|_2^2 - \|z^\star - z_{t+1}\|_2^2\right)\\ = \frac{\rho}{2}\sum_{i\in\mathcal{A}_{T-1}}\left(\|z^\star - z_0\|_2^2 - \|z^\star - z_T\|_2^2\right)\\ + \frac{\rho}{2}\sum_{i\notin\mathcal{A}_{T-1}}\left(\|z^\star - z_0\|_2^2 - \|z^\star - z_{\tilde{t}_i}\|_2^2\right) \qquad (19)\end{aligned}$$

Hence, from Assumption 1 *(d)* we have that

$$\frac{\rho}{2}\sum_{t=0}^{T-1}\sum_{i\in\mathcal{A}_t}\left(\|z^\star - z_{\bar{t}_i}\|_2^2 - \|z^\star - z_{t+1}\|_2^2\right) \leq \frac{N\rho D}{2} \qquad (20)$$

Now, from Assumption 1 *(b)* we know that $h(0) = 0$ and $h(z) \geq 0$ for all z, then we have the following lower bound

$$\begin{aligned}\sum_{t=0}^{T-1}\sum_{i\in\mathcal{A}_t} h(z_{t+1}) \geq h(0) - h(z_T) + \sum_{t=0}^{T-1}\sum_{i\in\mathcal{A}_t} h(z_{t+1})\\ = \sum_{t=0}^{T-1}\sum_{i\in\mathcal{A}_t} h(z_t) \qquad (21)\end{aligned}$$

Finally, by substituting (17), (18) and (20) into the telescopic sum over t of (12) and using (21) yields (13).

B. REFERENCES

[1] D. P. Palomar and Mung Chiang, “A tutorial on decomposition methods for network utility maximization,” *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1439–1451, 2006.

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2010.

[3] I.D. Schizas, A. Ribeiro, and G.B. Giannakis, “Consensus in ad hoc wsns with noisy links part i: Distributed estimation of deterministic signals,” *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 350–364, 2008.

[4] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Püschel, “D-ADMM: A communication-efficient distributed algorithm for separable optimization,” *IEEE Trans. Signal Process.*, vol. 61, no. 10, pp. 2718–2723, 2013.

[5] J. Matamoros, S. M. Fosson, Enrico Magli, and Carles Antón-Haro, “Distributed admm for in-network reconstruction of sparse signals with innovations,” *IEEE Trans. on Sign. Process. over Netws.*, vol. 1, pp. 224–234, 2015.

[6] Mingyi Hong, “A distributed, asynchronous and incremental algorithm for nonconvex optimization: An ADMM based approach,” *CoRR*, vol. abs/1412.6058, 2014, http://arxiv.org/abs/1412.6058.

[7] Tsung-Hui Chang, Mingyi Hong, Wei-Cheng Liao, and Xiangfeng Wang, “Asynchronous distributed ADMM for large-scale optimization- part I: algorithm and convergence analysis,” *CoRR*, vol. abs/1509.02597, 2015, http://arxiv.org/abs/1509.02597.

[8] Elad Hazan, “Introduction to online convex optimization,” 2015, http://ocobook.cs.princeton.edu/OCObook.pdf.

[9] Huahua Wang and Arindam Banerjee, “Online alternating direction method (longer version),” *CoRR*, vol. abs/1306.3721, 2013, http://arxiv.org/abs/1306.3721.

[10] M. Mahdavi, R. Jin, and M. Yang, T., “Trading regret for efficiency: Online convex optimization with long term constraints,” *Journal of Machine Learning Research*, vol. 13, pp. 2503–2528, 2012.