

COMMUNICATION-EFFICIENT WEIGHTED ADMM FOR DECENTRALIZED NETWORK OPTIMIZATION

Qing Ling^{*} Yaohua Liu^{*} Wei Shi[†] Zhi Tian[‡]

^{*} Department of Automation, University of Science and Technology of China

[†] Coordinated Science Laboratory, University of Illinois at Urbana-Champaign

[‡] Department of Electrical and Computer Engineering, George Mason University

ABSTRACT

In this paper, we propose a weighted alternating direction method of multipliers (ADMM) to solve the consensus optimization problem over a decentralized network. Compared with the conventional ADMM that is popular in decentralized network optimization, the weighted ADMM is able to tune its weight matrices for the purpose of reducing the communication cost spent in the optimization process. We first prove convergence and establish linear convergence rate of the weighted ADMM. Second, we maximize the derived convergence speed and obtain the best weight matrices on a given topology. Third, observing that exchanging information with all the neighbors is expensive, we maximize the convergence speed while limit the number of communication arcs. This strategy finds a subgraph within the underlying topology to fulfill the optimization task and leads to a favorable tradeoff between the number of iterations and the communication cost per iteration. Numerical experiments demonstrate advantages of the weighted ADMM over its conventional counterpart in expediting the convergence speed and reducing the communication cost.

Acknowledgement. Qing Ling and Yaohua Liu are supported by NSF China grant 61573331 and NSF Anhui grant 1608085QF130.

1. INTRODUCTION

Along with the rapid progress of data acquisition, communications and networking technologies, information processing and decision making over decentralized networks have attracted noticeable research interest in these years. A group of geographically distributed nodes, which are equipped with sensing, communicating and computing abilities, collaboratively accomplish an information processing or decision making task over an underlying network topology. A typical task is decentralized consensus optimization, in which n nodes solves

$$\min_x \sum_{i=1}^n f_i(x). \quad (1)$$

Here $x \in \mathcal{R}^p$ is the common optimization variable and $f_i : \mathcal{R}^p \rightarrow \mathcal{R}$ is the local objective function of node i . Such a problem formulation appears in various applications, for example, wireless communications and networking [1, 2], spectrum sensing of cognitive radios [3, 4], monitoring and optimization of smart grids [5, 6], distributed control of networked robots [7, 8, 9], to name a few.

In a decentralized algorithm that solves (1), every node holds its local objective function, exchanges current iterate with a subset of neighbors, carries on local computation, and eventually reaches an optimal solution that is consensual to all the nodes. In this optimization process, communication cost is one of the key consider-

ations of implementation. Reducing the amount of information exchange among the nodes alleviates bandwidth burden, improves system robustness, and enables fast information processing and decision making. In this paper, we propose a weighted alternating direction method of multipliers (ADMM) to solve (1), aiming at reducing the communication cost.

1.1. Related Works

Decentralized optimization algorithms that solve (1) include gradient/subgradient methods [10, 11] and their accelerated versions [12], diffusion methods [13, 14], dual averaging methods [15, 16], Newton methods [17, 18], and ADMM [1, 2, 19]. Among these algorithms, the decentralized ADMM has shown fast convergence in both practice and theory. When (1) is a convex program, ADMM converges to the optimal solution at a sublinear rate of $O(1/k)$ with k being the number of iterations [2]. Its linear rate of $O(\tau^k)$, where $\tau \in (0, 1)$ is a topology-dependent constant, is established in [20] given that the local objective functions are strongly convex. ADMM is also able to utilize the special composite structures or introduce surrogates of the local objective functions so as to significantly simplify the computation, while still keep its favorable convergence properties [21, 22, 23, 24].

Not surprisingly, convergence speed of the conventional decentralized ADMM is determined by condition numbers of the local objective functions, spectral properties of the underlying topology and stepsize of the dual gradient ascent step [20]. However, the conventional ADMM is unable to achieve the best communication efficiency due to two reasons. First, there is only one parameter, the ADMM stepsize, which can be tuned to maximize the convergence speed and consequently minimize the required number of iterations. Second, at every iteration, every node has to exchange its current iterate with all of its neighbors, which leads to a large amount of information exchange per iteration.

1.2. Our Contributions and Paper Organization

This paper proposes a weighted ADMM to solve the decentralized optimization problem (1) and address the two disadvantages of the conventional ADMM as discussed above. Intuitively, one can assign different weights to different nodes and arcs. Tuning the weights gives more flexibility to maximize the convergence speed than in the conventional ADMM. Furthermore, by setting some weights of arcs as zeros, we are able to avoid information exchange over a subset of arcs and hence reduce the communication cost per iteration.

Section 2 develops the weighted ADMM following this intuitive idea. Section 3 proves convergence and establishes linear rate of convergence for the weighted ADMM. We provide explicit expression of how the convergence speed is determined by the weight

Algorithm 1 Weighted ADMM run by node i **Require:** Initialize local iterates to $x_i^0 = 0$ and $\lambda_i^0 = 0$.1: **for** times $k = 0, 1, \dots$ **do**2: Compute local iterate x_i^{k+1} from

$$x_i^{k+1} = \arg \min_{x_i} f_i(x_i) + \langle x_i, \lambda_i^k - d_{ii}x_i^k + \sum_{j=1}^n a_{ij}x_j^k \rangle + d_{ii}\|x_i\|^2.$$

3: Transmit x_i^{k+1} and receive x_j^{k+1} from $j \in \mathcal{C}_i \subseteq \mathcal{N}_i$.4: Update local Lagrange multiplier λ_i^{k+1} as

$$\lambda_i^{k+1} = \lambda_i^k + d_{ii}x_i^{k+1} - \sum_{j=1}^n a_{ij}x_j^{k+1}.$$

5: **end for**

matrices, which enables optimal design of the latter. Section 4 gives two optimal design strategies. The first one simply maximizes the convergence speed, while the second one confines the number of communication arcs for the sake of reducing the amount of information exchange at every iteration. Numerical experiments in Section 5 demonstrate advantages of the weighted ADMM over its conventional counterpart in reducing the communication cost. Proofs of theorems, detailed discussions and extra numerical experiments are placed in a longer version of this paper [25].

2. ALGORITHM DEVELOPMENT**2.1. Problem Statement**

Network model. Consider a bidirectionally connected network consisting of n nodes and t edges. Describe the network as a symmetric directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where \mathcal{V} is the set of nodes with cardinality $|\mathcal{V}| = n$ and \mathcal{E} is the set of arcs with cardinality $|\mathcal{E}| = 2t$. Nodes i and j are neighbors of each other if $(i, j) \in \mathcal{E}$ and, by the symmetry of the network, $(j, i) \in \mathcal{E}$. The set of node i 's neighbors is denoted as \mathcal{N}_i , whose cardinality $|\mathcal{N}_i|$ is the degree of node i .

Communication model. At every iteration, every node i communicates with a set of other nodes \mathcal{C}_i , sending and receiving current local iterates. The communication is assumed to be synchronized. Furthermore, in order to guarantee that the algorithm is decentralized, every node is only allowed to communicate with those nodes in its neighbor set; namely, for every node i we must have $\mathcal{C}_i \subseteq \mathcal{N}_i$. Suppose that upon sending a message, every node i broadcasts once to all the nodes in \mathcal{C}_i and at every iteration. If every node encodes its local iterate with p bits, then the cost of sending is pn and the cost of receiving is $p \sum_{i=1}^n |\mathcal{C}_i|$ per iteration.

2.2. Weighted ADMM

In the weighted ADMM, every node i maintains a local variable $x_i \in \mathcal{R}^p$, which is a copy of the optimization variable x in (1). Node i also keeps a local variable $\lambda_i \in \mathcal{R}^p$, which plays the role of Lagrange multiplier as we will explain in Subsection 2.3. Both x_i and λ_i are updated using information collected from the nodes in \mathcal{C}_i . However, only x_i is transmitted to the nodes in \mathcal{C}_i ; λ_i is kept private.

Collect all local variables x_i and λ_i in two matrices

$$X \triangleq [x_1^T; \dots; x_n^T] \in \mathcal{R}^{n \times p}, \quad \Lambda \triangleq [\lambda_1^T; \dots; \lambda_n^T] \in \mathcal{R}^{n \times p}.$$

Define an aggregate objective function $f(X) = \sum_{i=1}^n f_i(x_i)$. The matrix form of the weighted ADMM update is given by

$$\begin{aligned} X^{k+1} &= \arg \min_X f(X) + \langle X, \Lambda^k - (D + A)X^k \rangle + \langle X, DX \rangle, \\ \Lambda^{k+1} &= \Lambda^k + (D - A)X^{k+1}. \end{aligned} \quad (2)$$

In (2), $D \in \mathcal{D}^{n \times n}$ is a diagonal matrix and its (i, i) -th element is positive and denoted by d_{ii} ; $A \in \mathcal{A}^{n \times n}$ is a symmetric matrix satisfying that its (i, j) -th element $a_{ij} = 0$ if nodes i and j are neither neighbors nor the same. Given a matrix A , define $\mathcal{C}_i = \{j | a_{ij} \neq 0, i \neq j\}$, which guarantees $\mathcal{C}_i \subseteq \mathcal{N}_i$.

Splitting the computation in the matrix form (2) to individual nodes, the update of node i is given by

$$\begin{aligned} x_i^{k+1} &= \arg \min_{x_i} f_i(x_i) + \langle x_i, \lambda_i^k - d_{ii}x_i^k - \sum_{j=1}^n a_{ij}x_j^k \rangle + d_{ii}\|x_i\|^2, \\ \lambda_i^{k+1} &= \lambda_i^k + d_{ii}x_i^{k+1} - \sum_{j=1}^n a_{ij}x_j^{k+1}. \end{aligned} \quad (3)$$

The algorithm can be implemented in a decentralized manner. In the update of x_i^{k+1} , node i needs to calculate the summation $\sum_{j=1}^n a_{ij}x_j^k$, which only requires the previous iterates x_i^k and x_j^k , $j \in \mathcal{C}_i$, as $a_{ij} = 0$ if $j \neq i$ and $j \notin \mathcal{C}_i$. The objective function $f_i(x_i)$ and the previous Lagrange multiplier λ_i^k are also locally available. Similarly, in the update of λ_i^{k+1} , agent i calculates the weighted summation $\sum_{j=1}^n a_{ij}x_j^{k+1}$ of the current local iterates; this can be done through communication with its neighbors. The weighted ADMM is outlined in Algorithm 1.

2.3. Connection of Weighted and Conventional ADMM

To see the connection between the proposed weighted ADMM and the conventional one, observe that [20] gives the matrix form of the conventional ADMM as

$$\begin{aligned} X^{k+1} &= \arg \min_X f(X) + \langle X, \Lambda^k - cUX^k \rangle + \langle X, c\frac{U+V}{2}X \rangle, \\ \Lambda^{k+1} &= \Lambda^k + cVX^{k+1}. \end{aligned} \quad (4)$$

Therein, c is the ADMM stepsize; U and V are the signless and signed Laplacian matrices of the network, respectively. Comparing (2) and (4), we can find that the conventional ADMM is a special case of the weighted ADMM through setting $D = c(U + V)/2$ and $A = c(U - V)/2$. Note that such choices of D and A satisfy the requirements of the weighted ADMM. In this case, D is the degree matrix whose (i, i) -th element d_{ii} denotes the degree of node i , while A is the incidence matrix whose (i, j) -th element a_{ij} equals to one if nodes i and j are connected and zero otherwise.

Extending the conventional ADMM to the weighted one is non-trivial. First, in the conventional ADMM, updating x_i and λ_i in node i involves communication with all the neighbors j in \mathcal{N}_i , because of the structures of U and V . Therefore, the conventional ADMM has a fixed communication cost of receiving $p \sum_{i=1}^n |\mathcal{N}_i|$ per iteration given the network topology. Contrarily, the weighted ADMM is able to reduce the communication cost of receiving by letting every node communicate with less neighbors; this can be done through wisely choosing the matrix A such that $\sum_{i=1}^n |\mathcal{C}_i|$ is less than $\sum_{i=1}^n |\mathcal{N}_i|$. Second, the conventional ADMM can optimize its convergence speed through tuning the stepsize c , since U and V are fixed given the network topology. Whereas, the weighted ADMM has the flexibility of tuning two matrices, D and A . Consequently, the weighted ADMM has the potential to achieve faster convergence speed than its conventional counterpart.

3. CONVERGENCE AND LINEAR RATE**3.1. Assumptions**

Unless otherwise stated, the convergence results in this section are given under Assumptions from 1 through 4. Assumptions 1 and 2

are basic, requiring that the underlying network to be connected and the solution set not null, respectively.

Assumption 1 (Network connectivity). *The network of n nodes is bidirectionally connected.*

Assumption 2 (Solution existence). *The solution set to (1), denoted by \mathcal{X}^* , is nonempty.*

Assumption 3 supposes that the local objective functions are differentiable and have Lipschitz continuous gradients. This assumption is necessary to prove convergence of the weighted ADMM. To further establish linear rate of convergence, we require the local objective functions to be strongly convex, as stated in Assumption 4.

Assumption 3 (Lipschitz continuous gradient). *The local objective functions f_i are proper closed convex, differentiable, and have Lipschitz continuous gradients. There is a positive constant $L_f > 0$ such that for any node i and for any pair of points x_a and x_b it holds $\|\nabla f_i(x_a) - \nabla f_i(x_b)\| \leq L_f \|x_a - x_b\|$.*

Assumption 4 (Strong convexity). *The local objective functions f_i are strongly convex. There is a positive constant $\mu_f > 0$ such that for any node i and for any pair of points x_a and x_b it holds $\langle x_a - x_b, \nabla f_i(x_a) - \nabla f_i(x_b) \rangle \geq \mu_f \|x_a - x_b\|^2$.*

3.2. Convergence Properties

In Theorem 1, we shall show that the sequence $\{X^k\}$ generated by the weighted ADMM converges to X^* , one of the optimal solutions. Here $X^* \triangleq [(x_1^*)^T; \dots; (x_n^*)^T] \in \mathbb{R}^{n \times p}$ is consensual, that is, $x_1^* = \dots = x_n^*$ and they are optimal to (1).

Theorem 1. *Under Assumptions 2 and 3 and given that the weighted matrices $D \in \mathcal{D}$ and $A \in \mathcal{A}$ are chosen such that $D + A \succeq 0$, $D - A \succeq 0$ and $\text{Null}(D - A) = e$ where $e = [1; \dots; 1]$ is an all-one vector, the sequence $\{X^k\}$ generated by the weighted ADMM converges to an optimal solution X^* .*

Provided that the local objective functions not only have Lipschitz continuous gradients but also are strongly convex, Theorem 2 further establishes linear rate of convergence. In particular, we obtain the convergence speed that is explicitly determined by the condition numbers of the local objective functions and the spectral properties of the weight matrices D and A .

Theorem 2. *Under Assumptions 2, 3 and 4 and given that the weighted matrices $D \in \mathcal{D}$ and $A \in \mathcal{A}$ are chosen such that $D + A \succeq 0$, $D - A \succeq 0$ and $\text{Null}(D - A) = e$, the sequence $\{X^k\}$ generated by the weighted ADMM converges at the linear rate $O((1 + \delta)^{-k})$ to the unique optimal solution X^* . Specifically, the convergence speed δ is any positive constant no larger than*

$$\min \left\{ \frac{\tilde{\sigma}_{\min}(D - A)}{\theta \sigma_{\max}(D + A)}, \frac{2\mu_f}{\sigma_{\max}(D + A) + \frac{\theta L_f^2}{(\theta - 1)\tilde{\sigma}_{\min}(D - A)}} \right\}. \quad (5)$$

Theorem 2 shows that the weighted ADMM converges linearly and its theoretically achievable speed is given by (5), which is determined by the Lipschitz gradient and strong convexity constants of the local objective functions (L_f and μ_f) and the spectral properties of the weight matrices D and A ($\sigma_{\max}(D + A)$ and $\tilde{\sigma}_{\min}(D - A)$). Suppose that the local objective functions are given a prior such that L_f and μ_f are fixed. The theoretically achievable speed is monotonically decreasing in $\sigma_{\max}(D + A)$ while increasing in $\tilde{\sigma}_{\min}(D - A)$. Hence, to accelerate the convergence speed and reduce the communication cost, we have the flexibility of tuning the weight matrices D and A so as to minimize $\sigma_{\max}(D + A)$ and maximize $\tilde{\sigma}_{\min}(D - A)$. Note that tuning the elements in D and A changes the weights of the individual nodes and arcs. This helps because in a given topology some nodes and arcs may contribute more to the information

diffusion process while the others contribute less. Intuitively, we expect to identify those important nodes and arcs and give them higher weights, which expedites propagation of “useful” information and reduces exchange of “less useful” messages.

4. MINIMIZING COMMUNICATION COST

This section investigates how to minimize the communication cost of the weighted ADMM through optimizing the spectral properties of the weight matrices D and A . Observe that the diagonal elements of D and A correspond to the nodes and the off-diagonal elements of A correspond to the arcs. If a_{ij} and a_{ji} are both zero and $i \neq j$, then nodes i and j have no information exchange even though there exists a communication link between nodes i and j . Therefore, given a predefined network topology $(\mathcal{V}, \mathcal{E})$, we propose two different strategies of tuning D and A . The first strategy allows every a_{ij} to be nonzero as long as $i \in \mathcal{N}_j$ (see Subsection 4.1). The second strategy lets some a_{ij} be zeros even though $i \in \mathcal{N}_j$, which is equivalent to selecting a subset of neighbors \mathcal{C}_j from \mathcal{N}_j and hence reduces the amount of information exchange per iteration (see Subsection 4.2).

4.1. Maximizing Speed

According to the theoretical analyses in Section 3, to maximize the convergence speed of the weighted ADMM through tuning the weight matrices D and A , the optimization model is

$$\begin{aligned} \min_{D, A} \quad & \{\sigma_{\max}(D + A), -\tilde{\sigma}_{\min}(D - A)\}, \\ \text{s.t.} \quad & D \in \mathcal{D}, A \in \mathcal{A}, D + A \succeq 0, D - A \succeq 0, \text{Null}(D - A) = e. \end{aligned} \quad (6)$$

However, the multi-objective optimization problem (6) is difficult to solve. Therefore, we propose an alternative approach that confines $\sigma_{\max}(D + A)$ to be less than a positive constant ρ while minimizes $-\tilde{\sigma}_{\min}(D - A)$. This way, we have a single-objective problem

$$\begin{aligned} \min_{D, A} \quad & -\tilde{\sigma}_{\min}(D - A), \\ \text{s.t.} \quad & D \in \mathcal{D}, A \in \mathcal{A}, D + A \succeq 0, D - A \succeq 0, \text{Null}(D - A) = e, \\ & \sigma_{\max}(D + A) \leq \rho. \end{aligned} \quad (7)$$

The optimization problem (6) is convex since the objective function and the set of constraints are both convex [26]. We solve (6) with CVX, a popular optimization toolbox [27].

4.2. Maximizing Speed Using Limited Communication Arcs

The overall communication cost of the weighted ADMM is determined by the product of the number of iterations and the communication cost per iteration. On a fixed topology, utilizing all the available communication arcs shall definitely achieve the fastest convergence speed, and hence reduce the number of iterations to reach a given accuracy. However, this strategy brings high communication cost per iteration. Indeed, some communication arcs are less important than the others and can be disconnected to reduce the amount of information exchange iteration-wise, as pointed out in Section 3. Therefore, in this subsection we propose an alternative strategy that maximizes the convergence speed under the constraint of limited communication arcs.

Observe that the number of communication arcs required in the weighted ADMM equals to the number of nonzero off-diagonal elements in A . Denote $\text{OffDiag}(A)$ as a matrix whose off-diagonal elements are identical to those of A and diagonal elements are zeros. Also denote the pseudo ℓ_0 norm $\|\text{OffDiag}(A)\|_0$ as the number of nonzero elements of $\text{OffDiag}(A)$. Suppose that we expect to use at

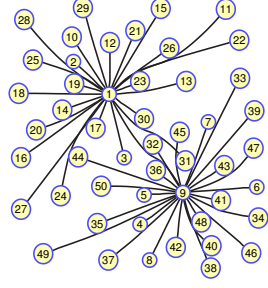


Fig. 1. A graph with two clusters of nodes.

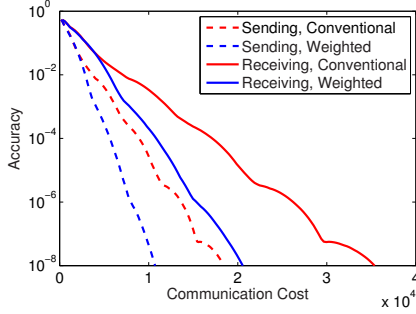


Fig. 2. Communication costs on the graph with two clusters.

most $2s$ arcs (namely, at most s edges due to the symmetry of A), the optimization of D and A turns to

$$\begin{aligned} \min_{(D, A) \in \Omega} \quad & -\tilde{\sigma}_{\min}(D - A), \\ \text{s.t.} \quad & \|\text{OffDiag}(A)\|_0 \leq 2s, \end{aligned} \quad (8)$$

where Ω is the feasible set of (7). The new formulation (8) is non-convex. We utilize ADMM to find a suboptimal solution of (8) because it has had successful applications in many optimization problems with ℓ_0 norm constraints [28, 29]. Observe that here ADMM is used to split the nonconvex constraint and the rest convex part, while in decentralized optimization ADMM is used to split the computation of the nodes. The algorithm to solve (8) is given in [25].

5. NUMERICAL EXPERIMENTS

This section compares the weighted and conventional ADMMs. We first show that through maximizing the convergence speed, the weighted ADMM achieves better communication efficiency than the conventional one on some graphs. The saving on the communication cost by the weighted ADMM becomes more significant when we maximize the convergence speed under the constraint of communication arcs. In the conventional ADMM, we hand-tune its stepsize to the optimal value. We let every node i has a local objective function $f_i(x) = (1/2)\|y_i - M_i x\|^2$, where $M_i \in \mathcal{R}^{m \times p}$ and $y_i \in \mathcal{R}^m$ for every i and their elements are generated following the standard normal distribution. In the numerical experiments, let $n = 50$ (number of the nodes), $p = 3$ (length of x_i), $m = 3$ (length of y_i), and $\rho = 1$ (scale of $\tilde{\sigma}_{\min}(D - A)$). Quality of the local iterates at time k is evaluated by accuracy, which is defined as $\max_i \|x_i^k - x^*\|$.

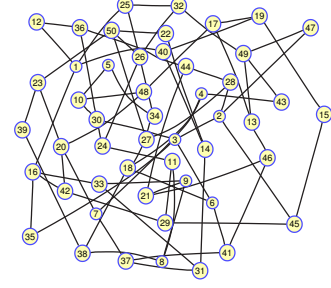


Fig. 3. The subgraph optimized by the weighted ADMM.

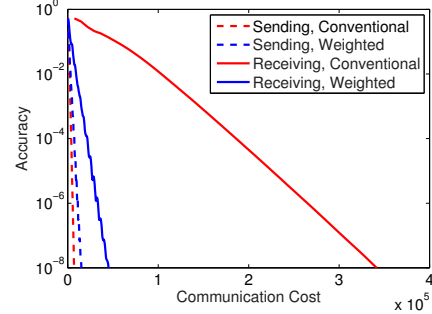


Fig. 4. Communication costs of the weighted ADMM on the optimized subgraph and the conventional one on the complete graph.

In the first experiment, we only maximize the convergence speed of the weighted ADMM. Interestingly, the conventional ADMM performs almost the same as the weighted one on most graphs, such as line, circle, star, complete, and random. But we observe that if the graph has several clusters of nodes (see Fig. 1 for an example of two clusters), then the weighted ADMM outperforms its conventional counterpart (see Fig. 2). This is reasonable because in the conventional ADMM, the cluster heads do not distinguish the neighboring ordinary nodes and the neighboring cluster heads. Through optimizing the weight matrices, the weighted ADMM properly emphasizes the importance of the cluster heads to their neighboring peers, and hence achieves better communication efficiency.

In the second experiment, we let the conventional ADMM run on a complete graph, but limit the number of communication arcs for the weighted ADMM. Optimally picking 150 arcs ($s = 75$ in (8)) out of 2450 possible ones, the resulting subgraph is given by Fig. 3. The communication costs of the two algorithms, in terms of sending and receiving, are demonstrated in Fig. 4. The conventional ADMM works on the complete graph so that its convergence is very fast, and consequently, the communication cost of sending is low. However, at every iteration every node must receive messages from all the other nodes, which is unaffordable in practice. As a comparison, the weighted ADMM provides a favorable tradeoff between the convergence speed and the communication cost per iteration. A noticeable byproduct of the weighted ADMM is that the selected subgraph is naturally load-balanced though we do not explicitly consider this metric in (8). Most of the nodes have 3 neighbors and some have 2 or 4, which is beneficial to the robustness of the network. We shall investigate this phenomenon in our future research.

6. REFERENCES

- [1] I. Schizas, A. Ribeiro, and G. Giannakis, "Consensus in ad hoc WSNs with noisy links - Part I: Distributed estimation of deterministic signals," *IEEE Transactions on Signal Processing*, vol. 56, pp. 350–364, 2008.
- [2] G. Giannakis, Q. Ling, G. Mateos, I. Schizas, and H. Zhu, "Decentralized learning for wireless communications and networking," Preprint at <http://arxiv.org/pdf/1503.08855v1.pdf>
- [3] F. Zeng, C. Li, and Z. Tian, "Distributed compressive spectrum sensing in cooperative multi-hop wideband cognitive networks," *IEEE Journal of Selected Topics on Signal Processing*, vol. 5, pp. 37–48, 2011.
- [4] J. Meng, W. Yin, H. Li, E. Hossain, and Z. Han, "Collaborative spectrum sensing from sparse observations in cognitive radio networks," *IEEE Journal of Selected Areas on Communications*, vol. 29, pp. 327–337, 2011.
- [5] G. Giannakis, N. Gatsis, V. Kekatos, S. Kim, H. Zhu, and B. Wollenberg, "Monitoring and optimization for power systems: A signal processing perspective," *IEEE Signal Processing Magazine*, vol. 30, pp. 107–128, 2013.
- [6] X. Li and A. Scaglione, "Robust decentralized state estimation and tracking for power systems via network gossiping," *IEEE Journal on Selected Areas in Communications*, vol. 31, pp. 1184–1194, 2013.
- [7] F. Bullo, J. Cortes, and S. Martinez, *Distributed Control of Robotic Networks*, Princeton University Press, 2009.
- [8] K. Zhou and S. Roumeliotis, "Multirobot active target tracking with combinations of relative observations," *IEEE Transactions on Robotics*, vol. 27, pp. 678–695, 2010.
- [9] Q. Ling and A. Ribeiro, "Decentralized dynamic optimization through the alternating direction method of multipliers," *IEEE Transactions on Signal Processing*, vol. 62, pp. 1185–1197, 2014.
- [10] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multiagent optimization," *IEEE Transactions on Automatic Control*, vol. 54, pp. 48–61, 2009.
- [11] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," Preprint at <http://arxiv.org/pdf/1310.7063v3.pdf>
- [12] D. Jakovetic, J. Xavier, and J. Moura, "Fast distributed gradient methods," *IEEE Transactions on Automatic Control*, vol. 59, pp. 1131–1146, 2014.
- [13] A. Sayed, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, pp. 311–801, 2014.
- [14] S. Tu and A. Sayed, "Distributed decision-making over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 62, pp. 1054–1069, 2014.
- [15] J. Duchi, A. Agarwal, and M. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Transactions on Automatic Control*, vol. 57, pp. 592–606, 2012.
- [16] K. Tsianos and M. Rabbat, "Distributed dual averaging for convex optimization under communication delays," In: *Proceedings of ACC*, 2012.
- [17] A. Mokhtari, Q. Ling, and A. Ribeiro, "An approximate Newton method for distributed optimization," In: *Proceedings of ICASSP*, 2015.
- [18] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, "DQM: Decentralized quadratically approximated alternating direction method of multipliers," Preprint at <http://arxiv.org/pdf/1508.02073v1.pdf>
- [19] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Second Edition, Athena Scientific, 1997.
- [20] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," *IEEE Transactions on Signal Processing*, vol. 62, pp. 1750–1761, 2014.
- [21] G. Mateos, J. Bazerque, and G. Giannakis, "Distributed sparse linear regression," *IEEE Transactions on Signal Processing*, vol. 58, pp. 5262–5276, 2010.
- [22] Q. Ling, W. Shi, G. Wu, and A. Ribeiro, "DLM: Decentralized linearized alternating direction method of multipliers," *IEEE Transactions on Signal Processing*, vol. 63, pp. 4051–4064, 2015.
- [23] T. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus ADMM," *IEEE Transactions on Signal Processing*, vol. 63, pp. 482–497, 2015.
- [24] P. Bianchi, W. Hachem, and F. Iutzeler, "A stochastic coordinate descent primal-dual algorithm and applications to large-scale composite optimization," Preprint at <http://arxiv.org/pdf/1407.0898v2.pdf>
- [25] Q. Ling, Y. Liu, W. Shi, and Z. Tian, "Communication-efficient weighted ADMM for decentralized network optimization," Preprint at <http://home.ustc.edu.cn/~qingling/pdf/WADMM.pdf>
- [26] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
- [27] M. Grant and S. Boyd, *CVX: Matlab Software for Disciplined Convex Programming*, Version 2.0 Beta, <http://cvxr.com/cvx>, 2013.
- [28] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, pp. 1–122, 2010.
- [29] G. Li and T. Pong, "Global convergence of splitting methods for nonconvex composite optimization," Preprint at <http://arxiv.org/pdf/1407.0753v5.pdf>