

RESEARCH

Open Access

# A multi-block alternating direction method with parallel splitting for decentralized consensus optimization

Qing Ling<sup>1\*</sup>, Min Tao<sup>2</sup>, Wotao Yin<sup>3</sup> and Xiaoming Yuan<sup>4</sup>

## Abstract

Decentralized optimization has attracted much research interest for resource-limited networked multi-agent systems in recent years. Decentralized *consensus* optimization, which is one of the decentralized optimization problems of great practical importance, minimizes an objective function that is the sum of the terms from individual agents over a set of variables on which all the agents should reach a consensus. This problem can be reformulated into an equivalent model with two blocks of variables, which can then be solved by the alternating direction method (ADM) with only communications between neighbor nodes. Motivated by a recently emerged class of so-called *multi-block* ADMs, this article demonstrates that it is more natural to reformulate a decentralized consensus optimization problem to one with multiple blocks of variables and solve it by a multi-block ADM. In particular, we focus on the multi-block ADM with parallel splitting, which has easy decentralized implementation. Convergence rate is analyzed in the setting of average consensus, and the relation between two-block and multi-block ADMs are studied. Numerical experiments demonstrate the effectiveness of the multi-block ADM with parallel splitting in terms of speed and communication cost and show that it has better network scalability.

## Introduction

In recent years, the communication, signal processing, control, and optimization communities have witnessed considerable research efforts on decentralized optimization for networked multi-agent systems [1-3]. A networked multi-agent system, such as a wireless sensor network (WSN) or a networked control system (NCS), is composed of multiple geographically distributed but interconnected agents which have sensing, computation, communication, and actuating abilities. This system generally has limited resources for communication, since battery power is limited and recharging is difficult, while communication between two agents is energy-consuming. Furthermore, the communication link is often vulnerable and bandwidth-limited. In this situation, decentralized optimization emerges as an effective approach to improve network scalability. In decentralized optimization, data and computation are decentralized. Each agent exchanges

information with its neighbors and accomplishes an otherwise centralized optimization task.

This article focuses on the *decentralized consensus optimization* problem. We consider a network of  $L$  agents which cooperatively optimize a separable objective function [3-8]:

$$\min \sum_{i=1}^L f_i(x), \quad (1)$$

where  $f_i(x) : \mathcal{R}^N \rightarrow \mathcal{R}$  is a convex function known to agent  $i$  only. The goal is to minimize the objective subject to consensus on  $x$ .

## Related study

The decentralized consensus optimization formulation (1) arises in many practical applications, such as averaging [9-11], estimation [12-17], learning [18-21], etc. The form of  $f_i(x)$  can be least squares [11-13],  $\ell_1$ -regularized least squares [14-17], or more general ones [18-21]. Note that this model can be extended to account for those with separable constraints, such as the network utility maximization (NUM) problem [22-24].

\*Correspondence: qingling@mail.ustc.edu.cn

<sup>1</sup>Department of Automation, University of Science and Technology of China, Hefei, Anhui, China

Full list of author information is available at the end of the article

Existing approaches to solving (1) include: i) belief propagation based on graphical models and Markovian random fields [18-20]; ii) incremental optimization which minimizes the overall objective function along a pre-defined path on the network [7,8]; iii) stochastic optimization with information exchange between neighboring agents [4-6]; and iv) optimization with explicit consensus constraints which can be handled with the alternating direction method (ADM) [3,12-17]. The ADM approach is fully decentralized, does not make any assumptions on network infrastructure such as free of loop or with a pre-defined path, and generally has satisfactory convergence performance. In this article, we mainly discuss the application of ADMs in the decentralized consensus optimization problem.

Our research is along the line of information-driven signal processing and control of WSNs and NCSs [24-26]. Accompanied with the unprecedented data collection abilities offered by large-scale networked multi-agent systems, a new challenge also arises: *how should we process such a large amount of data to make estimates and produce control strategies given limited network resources?* Instead of processing the data in a fusion center, our solution is letting each agent autonomously make decisions aided by limited communication with its neighbors. From this perspective, each individual objective function  $f_i(x)$  in (1) is constructed from the data collected by agent  $i$ , and  $x$  is the global information common to all agents (e.g., estimates or control strategies) obtained based on the data collected by the whole network. Though this framework can be generalized to various signal processing and control problems, this article focuses on those can be formulated as (1). For problems such as dynamic control and Kalman filtering of networked multi-agent systems, interested readers are referred to [1,2,27,28], respectively.

### Our contribution

Motivated by a series of recent articles on multi-block ADMs and their convergence analysis [29-31], this article describes their applications to the decentralized consensus optimization problem. The multi-block ADM with parallel splitting is reviewed in Section 3. Unlike the classical ADM (see textbooks [32,33]), this multi-block ADM splits the optimization variables into *multiple* blocks and sequentially updates just one of them while fixing the others. The classical ADM, on the other hand, only has two blocks of variables. Hence in this article we refer to it by the two-block ADM. Our problem (1) does not naturally have two distinct blocks of variables, and to apply the two-block ADM one needs to introduce extra variables (see e.g., [15,16,32]). We review this in Section 2. On the other hand, it is simpler to apply the multi-block ADM to (1) and the resulting algorithm is readily decentralized.

In this article also analyzes the convergence rate of the multi-block ADM applied to the average consensus problem, which is a special case of (1) where  $f_i(x) = \frac{1}{2} \|x - b_i\|_2^2$  for all  $i$ . In this setting, if the parameters of the multi-block ADM satisfy a certain formula, it is equivalent to the two-block ADM. Therefore, the two-block ADM can be considered as a special case of the multi-block ADM on average consensus problems. This relation also gives a guideline to select the parameters of the multi-block ADM so that it is not equivalent to and runs faster than the two-block ADM on all the tested decentralized consensus optimization problems, including the tested average consensus problems. The simulation results demonstrate that the multi-block ADM accelerates convergence, reduces communication cost, and thus improves network scalability.

### Paper organization

The rest of this article is organized as follows. Section 2 reviews a reformulation of the decentralized consensus optimization problem (1), to which the two-block ADM is applied. Section 3 reviews the multi-block ADM and applies a parallel-splitting version of it to (1). Section 4 elaborates on the convergence rate analysis on the average consensus problem, and shows that the two-block ADM is a special case of the multi-block ADM in this case. Section 5 presents numerical simulations of the two-block and multi-block ADMs. Finally, Section 6 concludes the article. Appendix 1 is placed in the last section.

### Problem formulation and the two-block ADM

In this section, we describe an equivalent formulation of the decentralized consensus optimization problem (1) and outline the algorithm design based on the two-block ADM.

#### Problem formulation

We consider a networked multi-agent system described by an undirected connected communication graph  $\mathcal{G} = (\mathcal{L}, \mathcal{E})$ , where  $\mathcal{L}$  is the set of  $L$  vertexes (distributed agents) and  $\mathcal{E}$  is the set of edges (communication links). There exists an edge  $(i, j) \in \mathcal{E}$  between agents  $i$  and  $j$  if they can directly communicate with each other. The two agents are also called one-hop neighbors, or simply neighbors. The set of one-hop neighbors of agent  $i$  is denoted by  $\mathcal{N}_i$ , whose cardinality is denoted by  $|\mathcal{N}_i|$ .

Our objective is to solve (1) with only information exchange between neighbors. To this end, define  $x^{(i)}$  as agent  $i$ 's local copy of  $x$  and impose consensus constraints  $x^{(i)} = x^{(j)}$  for all pairs of neighbors  $i$  and  $j$ . With these and given that the communication graph  $\mathcal{G}$  is connected, we obtain the following equivalent formulation of (1) (see e.g., [13]):

$$\begin{aligned} \min \quad & \sum_{i=1}^L f_i(x^{(i)}), \\ \text{s.t.} \quad & x^{(i)} = x^{(j)}, \quad \forall j \in \mathcal{N}_i, \forall i. \end{aligned} \quad (2)$$

### The two-block ADM

Let us consider the following convex program with separable equality constraints:

$$\begin{aligned} \min \quad & g_1(\theta_1) + g_2(\theta_2), \\ \text{s.t.} \quad & D_1\theta_1 + D_2\theta_2 = e. \end{aligned} \quad (3)$$

Here for  $i = 1$  and  $2$ ,  $g_i : \mathcal{R}^{N'_i} \rightarrow \mathcal{R}$  is convex,  $D_i \in \mathcal{R}^{M' \times N'_i}$ ,  $e \in \mathcal{R}^{M'}$ . The two-block ADM constructs the augmented Lagrangian function as:

$$\begin{aligned} L_a(\theta_1, \theta_2, \lambda) = & g_1(\theta_1) + g_2(\theta_2) + \lambda^T (D_1\theta_1 + D_2\theta_2 - e) \\ & + \frac{c}{2} \|D_1\theta_1 + D_2\theta_2 - e\|_2^2. \end{aligned}$$

Here  $\lambda \in \mathcal{R}^{M'}$  is a Lagrange multiplier and  $c$  is a positive constant. At the  $t$ th iteration, the two-block ADM updates the optimization variables  $\theta_1(t+1)$  and  $\theta_2(t+1)$  as:

$$\begin{aligned} \theta_1(t+1) &= \arg \min_{\theta_1} L_a(\theta_1, \theta_2(t), \lambda(t)), \\ \theta_2(t+1) &= \arg \min_{\theta_2} L_a(\theta_1(t+1), \theta_2, \lambda(t)), \end{aligned}$$

and updates the Lagrange multiplier  $\lambda(t+1)$  as:

$$\lambda(t+1) = \lambda(t) + c(D_1\theta_1(t+1) + D_2\theta_2(t+1) - e).$$

The two-block ADM guarantees global convergence for any  $c > 0$  [32]. More precisely, when each  $g_i$  is convex for  $i = 1$  and  $2$ , the dual sequence  $\{\lambda(t)\}$  converges to an optimal dual solution of (5); if further the primal sequence  $\{[\theta_1(t)^T, \theta_2(t)^T]^T\}$  is bounded, the sequence converges to an optimal primal solution of (5).

### The two-block ADM for decentralized consensus optimization

The two-block ADM cannot be directly applied to problem (2) because its constraints interconnect all the variables pair by pair. There are no obvious two blocks. To overcome this, [32] describes a new block of auxiliary variables, and reformulates (2) as:

$$\begin{aligned} \min_{\{x^{(i)}, \{z_{ij}\}\}} \quad & \sum_{i=1}^L f_i(x^{(i)}), \\ \text{s.t.} \quad & x^{(i)} = z_{ij}, x^{(j)} = z_{ij}, \forall j \in \mathcal{N}_i, \forall i. \end{aligned} \quad (4)$$

Here  $z_{ij}$  is an auxiliary variable attached to  $x^{(i)}$  and  $x^{(j)}$ .

Treating  $\{x^{(i)}\}$  and  $\{z_{ij}\}$  as two blocks of variables, the two-block ADM is applied to problem (4). This technique has been adopted in [15,16] to solve the decentralized consensus optimization problem with neighboring consensus constraints. After eliminating  $\{z_{ij}\}$  from the iterative

updates and further simplifications, the two-block ADM for (4) is given below as algorithm TB-ADM.

**Initialization:** Each agent  $i$  initializes  $x^{(i)}(0) = 0$  and  $\alpha_i(0) = 0$ .

**Step 1:** At time  $t$ , each agent  $i$  updates its local copy  $x^{(i)}$  as:  $x^{(i)}(t+1) = \arg \min_{x^{(i)}} f_i(x^{(i)}) + \alpha_i^T(t)x^{(i)} + c \sum_{j \in \mathcal{N}_i} \|x^{(i)} - \frac{1}{2}(x^{(i)}(t) + x^{(j)}(t))\|_2^2$ , where  $\alpha_i$  is the Lagrange multiplier and  $c$  is a positive constant.

**Step 2:** At time  $t$ , each agent  $i$  updates its Lagrange multiplier  $\alpha_i$  as:

$$\alpha_i(t+1) = \alpha_i(t) + c|\mathcal{N}_i|x^{(i)}(t+1) - c \sum_{j \in \mathcal{N}_i} x^{(j)}(t+1).$$

**Step 3:** Repeat Step 1 and Step 2 until convergence.

TB-ADM is well suited for decentralized computation since the updates require only communication between agents  $i$  and  $j$ , who are one-hop neighbors. Detailed derivation of TB-ADM can be found in [15,16,32].

### The multi-block ADM

The fact that many practical optimization problems naturally have multiple blocks of variables motivates the development of a class of multi-block ADMs, such as the one with parallel splitting [29], with prediction-correction [30], and with Gaussian back substitution [31]. Due to the nature of the decentralized consensus optimization problem (2) and the need of parallelization, we choose the multi-block ADM with parallel splitting in [29].

### The multi-block ADM with parallel splitting

Consider an equality constrained convex program which can be separated to  $L$  parts:

$$\begin{aligned} \min \quad & \sum_{i=1}^L g_i(\theta_i), \\ \text{s.t.} \quad & \sum_{i=1}^L D_i\theta_i = e. \end{aligned} \quad (5)$$

Here for all  $i$ ,  $g_i : \mathcal{R}^{N'_i} \rightarrow \mathcal{R}$  is convex,  $D_i \in \mathcal{R}^{M' \times N'_i}$ ,  $e \in \mathcal{R}^{M'}$ . At the  $t$ th iteration, the multi-block ADM with parallel splitting works as follows:

*Step 1: Updating an auxiliary variable  $q$ :*

$$q(t+1) = \lambda(t) + \beta \left( \sum_{i=1}^L D_i\theta_i(t) - e \right),$$

where  $q$  is an auxiliary variable,  $\lambda$  is a Lagrange multiplier, and  $\beta$  is a positive constant.

*Step 2: Updating optimization variables  $\{\theta_i\}$ :*

$$\theta_i(t+1) = \arg \min_{\theta_i} g_i(\theta_i) + q(t+1)^T D_i \theta_i + \frac{\mu}{2} \|D_i \theta_i - D_i \theta_i(t)\|_2^2, \quad \forall i,$$

where  $\mu$  is a positive constant.

*Step 3: Updating the Lagrange multiplier  $\lambda$ :*

$$\lambda(t+1) = \lambda(t) + \beta \left( \sum_{i=1}^L D_i \theta_i(t+1) - e \right).$$

The multi-block ADM guarantees global convergence if the two positive constants  $\beta$  and  $\mu$  are properly chosen. For the convergence proof and the settings of  $\beta$  and  $\mu$ , the interested reader is referred to [29].

### The multi-block ADM for decentralized consensus optimization

Applying the multi-block ADM in (2) directly gets a decentralized algorithm, and does not need to introduce a new block of auxiliary variables and eliminate them, as we have done in the two-block ADM. We provide the algorithm to solve (2) based on the multi-block ADM with parallel splitting, denoted as MB-ADM. Detailed derivation of MB-ADM is given in Appendix 1.

**Initialization:** Each agent  $i$  initializes  $q_i(0) = 0$ ,  $x^{(i)}(0) = 0$ , and  $\lambda_i(0) = 0$ .

**Step 1:** At time  $t$ , each agent  $i$  updates its auxiliary variable  $q_i$  as:

$$q_i(t+1) = \lambda_i(t) + \beta |\mathcal{N}_i| x^{(i)}(t) - \beta \sum_{j \in \mathcal{N}_i} x^{(j)}(t), \text{ where}$$

$\beta$  is a positive constant.

**Step 2:** At time  $t$ , each agent  $i$  updates its local copy  $x^{(i)}$  as:  $x^{(i)}(t+1) = \arg \min_{x^{(i)}} f_i(x^{(i)}) + 2q_i^T(t+1)x^{(i)} + \mu |\mathcal{N}_i| \|x^{(i)} - x^{(i)}(t)\|_2^2$ , where  $\mu$  is a positive constant.

**Step 3:** At time  $t$ , each agent  $i$  updates its Lagrange multiplier  $\lambda_i$  as:

$$\lambda_i(t+1) = \lambda_i(t) + \beta |\mathcal{N}_i| x^{(i)}(t+1) - \beta \sum_{j \in \mathcal{N}_i} x^{(j)}(t+1).$$

**Step 4:** Repeat Step 1 to Step 3 until convergence.

In each iteration, to update  $q_i(t+1)$  and  $\lambda_i(t)$ , agent  $i$  needs  $x^{(j)}(t)$  with the size of  $N \times 1$  from all neighbors  $j \in \mathcal{N}_i$ ; to optimize  $x^{(i)}(t+1)$ , agent  $i$  only needs local information  $q_i(t+1)$  and  $x^{(i)}(t)$ . In all, each agent only needs to broadcast an  $N \times 1$  vector of its local copy (i.e.,  $x^{(i)}(t)$ ) to its neighbors per iteration. MB-ADM and TB-ADM have the same per-iteration communication cost. At the  $t$ th iteration, agent  $i$  needs to update  $x^{(i)}(t)$ ,  $q_i(t)$ , and  $\lambda_i(t)$  in its memory for MB-ADM. Hence the memory requirement is slightly higher than that of TB-ADM, for which only  $x^{(i)}(t)$  and  $\alpha_i(t)$  need to be updated.

### Convergence rate analysis

Convergence rate is an significant issue for decentralized algorithms, since it directly influences the overall communication cost. With respect to general separable convex programs, [29,34] proves the sublinear convergence rates of  $\sim \frac{1}{t}$  for the multi-block and two-block ADMs, respectively. However, when they are applied to the average consensus problems, much faster convergence can be observed. For this reason, we improve the convergence rate in this section.

The average consensus problem gives rise to problem (2) with  $f_i(x^{(i)}) = \frac{1}{2} \|x^{(i)} - b_i\|_2^2, \forall i$  [9-11]; namely, agents aims at averaging their original measurements  $\{b_i\}$  via one-hop communication. Without loss of generality, we assume that  $x^{(i)}$  and  $b_i$  are both scalars since their dimensions have no effect on the convergence rate.

### Convergence rate of MB-ADM

In analyzing the convergence rate of MB-ADM for the average consensus problem, we first rewrite MB-ADM as a state transition equation form and then use the spectral analysis tools to provide a bound of convergence rate. Our train of thought is similar to that in [35] for the two-block ADM.

According to the derivation in Appendix 1, we can rewrite MB-ADM in a state transition equation form. Let us define a state vector  $s_M(t+1) = [x^{(1)}(t+1), \dots, x^{(L)}(t+1), x^{(1)}(t), \dots, x^{(L)}(t)]^T$  and the corresponding state transition equation of MB-ADM is:

$$s_M(t+1) = \Phi_M s_M(t). \quad (6)$$

Here the state transition matrix  $\Phi_M$  is defined as:

$$\Phi_M = \begin{pmatrix} \Gamma_M & \Omega_M \\ I_{L \times L} & 0_{L \times L} \end{pmatrix}$$

with  $\Gamma_M$  being an  $L \times L$  matrix whose  $(i, i)$ th entry is  $\frac{1-4\beta|\mathcal{N}_i|+4\mu|\mathcal{N}_i|}{1+2\mu|\mathcal{N}_i|}$  and  $(i, j)$ th entry is  $\frac{4\beta}{1+2\mu|\mathcal{N}_i|}$  if  $i$  and  $j$  are neighbors, and  $\Omega_M$  being an  $L \times L$  matrix whose  $(i, i)$ th entry is  $\frac{2\beta|\mathcal{N}_i|-2\mu|\mathcal{N}_i|}{1+2\mu|\mathcal{N}_i|}$  and  $(i, j)$ th entry is  $-\frac{2\beta}{1+2\mu|\mathcal{N}_i|}$  if  $i$  and  $j$  are neighbors. We can see that summation of each row of  $\Phi_M$  is 1. The initial state is  $s_M(1) = [\frac{b_1}{1+2\mu|\mathcal{N}_1|}, \dots, \frac{b_L}{1+2\mu|\mathcal{N}_L|}, 0, \dots, 0]$  when each agent  $i$  initializes  $q_i(0) = 0$ ,  $x^{(i)}(0) = 0$ , and  $\lambda_i(0) = 0$ .

**Proposition 1. (convergence and convergence rate of MB-ADM on average consensus)** The state transition equation (6) defined above has the following properties:

**Property 1.** The matrix  $\Phi_M$  has an eigenvalue  $\rho_{M1} = 1$  with multiplicity 1, and its corresponding left and right eigenvectors are:

$$l_{M1} = [1 + 2\mu|\mathcal{N}_1|, \dots, 1 + 2\mu|\mathcal{N}_L|, -2\mu|\mathcal{N}_1|, \dots, -2\mu|\mathcal{N}_L|]$$

and

$$r_{M1} = \left[ \frac{1}{L}, \dots, \frac{1}{L}, \frac{1}{L}, \dots, \frac{1}{L} \right]^T,$$

respectively. Note that  $l_{M1}$  and  $r_{M1}$  are chosen subject to  $l_{M1}r_{M1} = 1$ .

**Property 2.** Define:

$$\rho_M = \max_{i \neq 1} |\rho_{Mi}|,$$

where  $\rho_{Mi}$  is the  $i$ th eigenvalue of  $\Phi_M$ . If  $\rho_M < 1$ , then the limit property of  $s_M(t)$  is:

$$\begin{aligned} \lim_{t \rightarrow \infty} s_M(t+1) &= \lim_{t \rightarrow \infty} \Phi_M^t s_M(1) = r_{M1} l_{M1} s_M(1) \\ &= \frac{1}{L} \sum_{i=1}^L b_i [1, \dots, 1, 1, \dots, 1]^T. \end{aligned}$$

Further, denoting that  $\kappa_M$  is the size of the largest Jordan block of  $\Phi_M$ , the convergence rate is:

$$\|s_M(t+1) - s_M(\infty)\|_2 \sim t^{(\kappa_M-1)} \rho_M^t.$$

Proof of Property 1 is given in Appendix 1. Property 2 comes from the classical convergence rate analysis of state transition equations. If  $\rho_{M1} = 1$  and  $\rho_M < 1$ , then there exists a unique  $s_M(\infty)$  and the convergence rate is  $\|s_M(t+1) - s_M(\infty)\|_2 \sim t^{(\kappa_M-1)} \rho_M^t$  (see [36], Fact 3). Next we try to find one possible (and hence unique)  $s_M(\infty)$ . By definition,  $\Phi_M r_{M1} = \rho_{M1} r_{M1} = r_{M1}$ . Hence  $\lim_{t \rightarrow \infty} \Phi_M^t r_{M1} = r_{M1}$ . Similarly,  $\lim_{t \rightarrow \infty} l_{M1} \Phi_M = l_{M1}$ . These two facts mean that  $r_{M1} l_{M1}$  is a possible limit point of  $\lim_{t \rightarrow \infty} \Phi_M^t$ . Therefore,  $r_{M1} l_{M1} s_M(1)$  is a possible (and hence unique) limit point of  $s_M(\infty)$ .

**Remark 1.** Note that the  $\sim t^{(\kappa_M-1)} \rho_M^t$  rate, though still loose, is tighter than the  $\sim \frac{1}{t}$  rate of the multi-block ADM for general separable convex programs [29]. Indeed, from numerical experiments, we find that  $\kappa_M$ , the size of the largest Jordan block of  $\Phi_M$ , is often equal to 1 (it means that  $\Phi_M$  is diagonalizable). In this case, the convergence rate can be as fast as  $\sim \rho_M^t$ .

In Property 2, there is a condition that  $\rho_M < 1$ . It is not necessarily for true any choices of  $\mu$  and  $\beta$ . Next we show two nontrivial special cases where the condition in Property 2 satisfy. The first special case connects MB-ADM with TB-ADM. Analysis of these two special cases as well as numerical simulations provide guidelines for parameter selection in MB-ADM.

**Proposition 2. (two nontrivial special cases)** We have  $\rho_m < 1$  in either one of the following two cases:

Case 1: The parameters  $\mu$  and  $\beta$  are chosen such that  $\mu = 2\beta > 0$ ; further,  $\frac{2\beta|\mathcal{N}_j|}{1+2\mu|\mathcal{N}_j|} < \frac{1}{4}$  and  $\frac{2\mu|\mathcal{N}_j|}{1+2\mu|\mathcal{N}_j|} < \frac{1}{2}$  for all  $j = 1, 2, \dots, L$ .

Case 2: The parameters  $\mu$  and  $\beta$  are chosen such that  $\mu = \beta > 0$ ; further,  $\frac{2\beta|\mathcal{N}_j|}{1+2\mu|\mathcal{N}_j|} < \frac{1}{2}$  and  $\frac{2\mu|\mathcal{N}_j|}{1+2\mu|\mathcal{N}_j|} < \frac{1}{2}$  for all  $j = 1, 2, \dots, L$ .

**Remark 2.** The proof of Proposition 2 is given in Appendix 1. In case 1, we set  $\mu = 2\beta > 0$ , which indeed leads to the equivalence between MB-ADM and TB-ADM, as we will show in the next subsection. In case 2, we set  $\mu = \beta > 0$ , which brings faster convergence for the average consensus problem according to numerical simulations (see Section 5.2). Hence we recommend to set  $\beta = \tau\mu$  with a fixed ratio  $\frac{1}{2} \leq \tau \leq 1$ , and just tune the value of  $\mu$ . This setting also works well for the general decentralized consensus optimization problem (1). Tuning  $\mu$  for MB-ADM is similar to tuning  $c$  for TB-ADM; both algorithms have 1 parameter subject to the user choice. Note that the conditions in Proposition 2 are merely sufficient;  $\frac{2\beta|\mathcal{N}_j|}{1+2\mu|\mathcal{N}_j|}$  and  $\frac{2\mu|\mathcal{N}_j|}{1+2\mu|\mathcal{N}_j|}$  can be larger than their upper bounds given above.

#### Connection between MB-ADM and TB-ADM

To show the connection between MB-ADM and TB-ADM, we also write TB-ADM as a state transition equation form. Note that [35] considers another kind of two-block ADM for the average consensus problem, where consensus constraints are quadratically penalized by different weights in the augmented Lagrangian function. In TB-ADM, the consensus constraints are quadratically penalized by the same weight  $c$ .

We define a state vector  $s_T(t+1) = [x^{(1)}(t+1), \dots, x^{(L)}(t+1), x^{(1)}(t), \dots, x^{(L)}(t)]^T$  and the corresponding state transition equation, according to the derivation in Appendix 1:

$$s_T(t+1) = \Phi_T s_T(t). \quad (7)$$

Here the state transition matrix  $\Phi_T$  is defined as:

$$\Phi_T = \begin{pmatrix} \Gamma_T & \Omega_T \\ I_{L \times L} & 0_{L \times L} \end{pmatrix}$$

with  $I_{L \times L}$  being the  $L \times L$  identity matrix,  $0_{L \times L}$  being the  $L \times L$  zero matrix,  $\Gamma_T$  being an  $L \times L$  matrix whose  $(i, i)$ th entry is 1 and  $(i, j)$ th entry is  $\frac{2c}{1+2c|\mathcal{N}_i|}$  if  $i$  and  $j$  are neighbors, and  $\Omega_T$  being an  $L \times L$  matrix whose  $(i, i)$ th entry is  $-\frac{c|\mathcal{N}_i|}{1+2c|\mathcal{N}_i|}$  and  $(i, j)$ th entry is  $-\frac{c}{1+2c|\mathcal{N}_j|}$  if  $i$  and  $j$  are neighbors. The initial state is  $s_T(1) = [\frac{b_1}{1+2c|\mathcal{N}_1|}, \dots, \frac{b_L}{1+2c|\mathcal{N}_L|}, 0, \dots, 0]$  when each agent  $i$  initializes  $x^{(i)}(0) = 0$  and  $\alpha_i(0) = 0$ .

Comparing the state transition equations of MB-ADM and TB-ADM, we can find that TB-ADM is indeed a special case of MB-ADM when  $c = \mu = 2\beta > 0$ . In this sense, MB-ADM provides more flexibility in parameter selection than TB-ADM. According to our simulations in Section 5.2, setting  $\beta = \tau\mu$  with  $\frac{1}{2} \leq \tau \leq 1$  makes MB-ADM faster than TB-ADM.

Let  $\rho_{Ti}$  be the  $i$ th eigenvalue of  $\Phi_T$ . Apparently  $\rho_{T1} = 1$ . Defining:

$$\rho_T = \max_{i \neq 1} |\rho_{Ti}|,$$

and denoting  $\kappa_T$  as the size of the largest Jordan block of  $\Phi_T$ , we can prove that TB-ADM has a similar  $\sim t^{(\kappa_T-1)}\rho_T^t$  convergence rate to the optimal solution given the conditions in Case 1 of Proposition 2. Interestingly, the upper bounds of  $\frac{2c|\mathcal{N}_j|}{1+2c|\mathcal{N}_j|} < \frac{1}{2}$  for all  $j = 1, 2, \dots, L$  are no longer needed since TB-ADM guarantees global convergence for any  $c > 0$ .

### Numerical Experiments

In this section, we present numerical simulations and demonstrate the performance of MB-ADM on the decentralized consensus optimization problems. Particularly, we are interested in how the communication cost scales to the network size.

#### Simulation Settings

In the numerical experiments, we consider the case that the agents cooperatively solve a least-squares problem. Each agent  $i$  has a measurement matrix  $A_i \in \mathcal{R}^{M \times N}$  and a measurement vector  $b_i \in \mathcal{R}^M$ . The objective function in (1) is thus  $f(x) = \sum_{i=1}^L f_i(x) = \frac{1}{2} \sum_{i=1}^L \|A_i x - b_i\|_2^2$ . The elements of the true signal vector  $x_0$  and the entries of the measurement matrices  $\{A_i\}$  follow the normal distribution  $\mathcal{N}(0, 1)$ . The measurement vector  $b_i = A_i x_0 + \eta_i$ ; the elements of the noise vector  $\eta_i$  follow the normal distribution  $\mathcal{N}(0, 0.1)$ . In the tests of average consensus,  $\{A_i\}$  reduce to identity matrices and are no longer random.

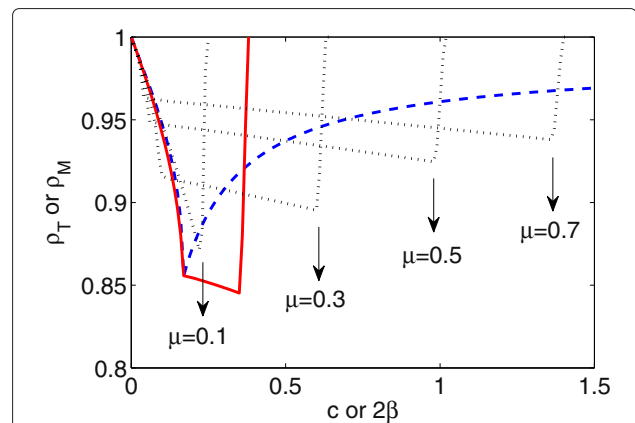
In the simulation, we assume that  $L$  agents are uniformly randomly deployed in a  $100 \times 100$  area. All agents have a common communication range  $r_C$ , which is chosen such that the networked multi-agent system is connected. Given  $r_C$ , the average node degree  $d$  can be calculated. We consider the following three scenarios: #1)  $L = 50, M = 1, N = 1, \{A_i = 1\}, r_C = 30, d \simeq 12$ ; #2)  $L = 50, M = 10, N = 5, r_C = 30, d \simeq 12$ ; #3)  $L = 200, M = 10, N = 5, r_C = 15, d \simeq 12$ . Scenario #1 is the average consensus test. Throughout the simulations, we set  $\beta = \tau\mu$  in MB-ADM with  $\tau = 0.9$ .

#### Convergence rate for average consensus

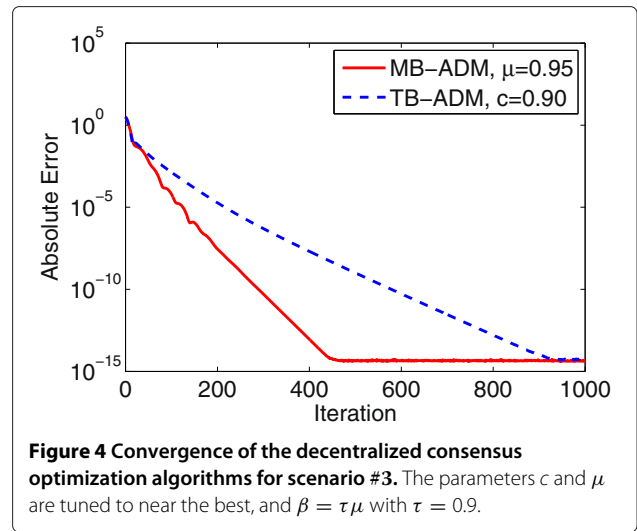
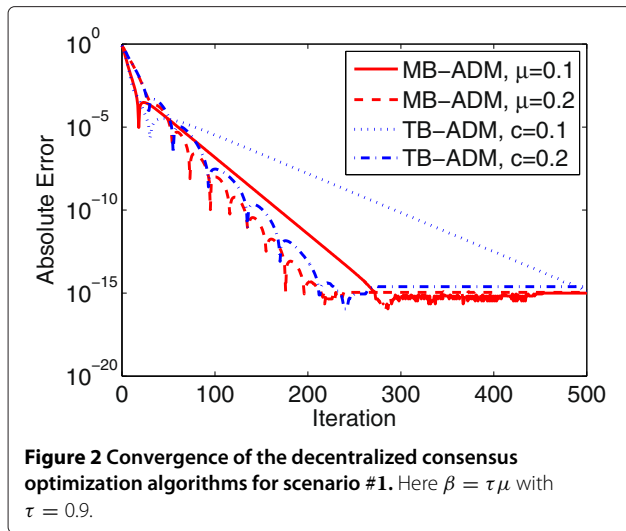
Under different choices of  $c, \mu$ , and  $\beta$ , the values of  $\rho_T$  for TB-ADM and the values of  $\rho_M$  for MB-ADM with respect to scenario #1 are shown in Figure 1. For TB-ADM,  $\rho_T$  sharply reduces when  $c$  increases from 0; after a certain turning point (at  $c^* \simeq 0.17$ ) which corresponds to the fastest convergence rate,  $\rho_T$  steadily increases. The curve of  $\rho_M$  for MB-ADM shows to be more complicated due to the existence of two parameters,  $\mu$  and  $\beta$ . For each  $\mu$ ,  $\rho_M$  steadily reduces when  $\beta$  increases from 0, then sharply goes to be larger than 1 which corresponds to divergence. The larger  $\mu$ , the wider convergence range for  $\beta$ ; but the side-effect is the relatively slower convergence rate. The curve of particular interest to us is  $\mu = c^* \simeq 0.17$ . In this curve,  $2\beta = c^* \simeq 0.17$  corresponds to Case 1 in Proposition 2; namely, when MB-ADM reduces to TB-ADM. Increasing  $\beta$  from  $c^*$ ,  $\rho_M$  still decreases until reaching a turning point  $2\beta = 2c^* \simeq 0.34$ , which corresponds to Case 2 in Proposition 2. This simulation validates our analysis in Section 5.2, as well as the proposed parameter selection rule (namely, setting a ratio  $\tau, \frac{1}{2} \leq \tau \leq 1$ , such that  $\beta = \tau\mu$ ).

Simulation results about the actual convergence properties are shown in Figure 2. By absolute error we denote the  $\ell_2$ -norm of the distance between the current solution and the centralized optimal solution. Though the convergence rates of MB-ADM and TB-ADM are at the same magnitude, MB-ADM shows to be slightly superior to TB-ADM.

According to the theoretical analysis in Sections 4.1 and 4.2, the estimated convergence rates of MB-ADM and TB-ADM are  $\sim t^{(\kappa_M-1)}\rho_M^t$  and  $\sim t^{(\kappa_T-1)}\rho_T^t$ , respectively. However, numerical simulations show that they are loose bounds; the actual convergence rate, as we can observe from Figure 2, are linear.



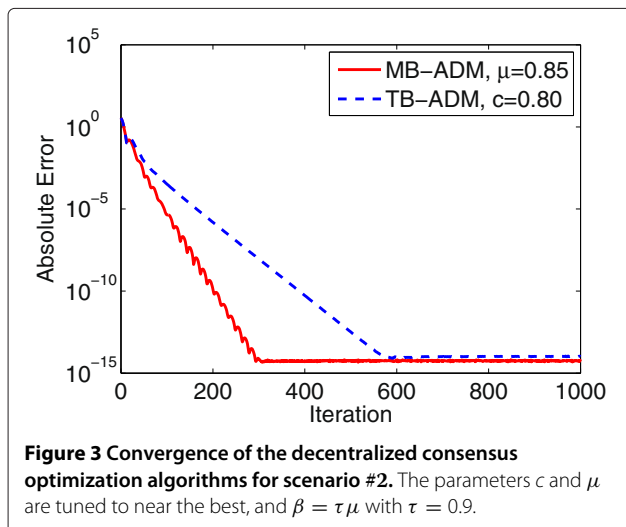
**Figure 1** Curves of  $\rho_T$  and  $\rho_M$  for TB-ADM MB-ADM in scenario #1. The dash line is for TB-ADM and its turning point corresponds to  $c^* \simeq 0.17$ ; the solid line is for MB-ADM with  $\mu = 2c^* \simeq 0.34$ ; the four dot lines are for MB-ADM with  $\mu = 0.1, 0.3, 0.5$ , and  $0.7$ .



**Performance Comparison**

Figures 3 and 4 depict the convergence properties of the two decentralized consensus optimization algorithms for scenarios #2 and #3, respectively. The parameters  $\mu$  and  $c$  are tune to be near the best ones with. Here we still have  $\beta = \tau\mu$  with  $\tau = 0.9$ . For either the medium network in scenario #2 or the large network in scenario #3, both algorithms linearly converge to the optimal solution. Comparing the two decentralized algorithms, MB-ADM outperforms TB-ADM in each scenario regarding convergence rate.

What of particular interest to us is whether the decentralized algorithms are scalable to network size. Observing Figure 3 with  $L = 50$  agents and  $d \simeq 12$ , and Figure 4 with  $L = 200$  agents and  $d \simeq 12$ , we can find that the convergence rates of the two algorithm are more dependent on the average node degree other than on the network size.



These numerical experiments verify the well-recognized claim that decentralized optimization may improve the performance of a networked multi-agent system with respect to network scalability.

**Communication cost**

Communication cost, in terms of energy consumption and bandwidth, is the major design consideration of a resource-limited networked multi-agent system, and can be approximately evaluated by the volume of information exchange during the decentralized consensus optimization process. Ignoring the extra burden of coordinating the network, for each agent, the communication cost is proportional to the number of iterations multiplied by the volume of information exchange per iteration. Therefore, reducing the information exchange per iteration is of critical importance to the design of lightweight algorithms.

Comparing the two decentralized consensus optimization algorithms, the information exchange per iteration is decided by the communication mode of agents, namely, broadcast or unicast. In the broadcast mode, one agent can send one piece of information to all of its neighbors with one transmission; contrarily, in the unicast mode, the agent needs multiple transmissions to do so. The two modes both have their pros and cons. The broadcast mode utilizes the characteristic of wireless communication, but may brings difficulties in coordinating the network, such as avoiding collisions. Though the unicast mode consumes much more transmissions, the randomized-gossip-like scheme is very useful in communication for the sake of robustness [37]. The average volume of information exchange per iteration of the four decentralized consensus optimization algorithms are outlined in Table 1, for both the broadcast and unicast modes.

In summary, the decentralized consensus optimization algorithms, no matter with the broadcast or unicast mode,

**Table 1 Average volume of information exchange per iteration**

	TB-ADM	MB-ADM
Broadcast mode	$N$	$N$
Unicast mode	$dN$	$dN$

are scalable to the network size. Since the number of iterations is proportional to the average node degree  $d$ , the overall average volume of information exchange is  $\sim Nd$  for the broadcast mode and  $\sim Nd^2$  for the unicast mode. As a comparison, consider a centralized networked multi-agent system uniformly randomly deployed in a two-dimensional area with a fusion center which collects measurement vectors from all agents. The average volume of information exchange is  $\sim M\sqrt{L}$  while the worst one is  $\sim ML$  for agents near the fusion center. When the network size  $L$  increases, the communication cost caused by the centralized network infrastructure is unaffordable and the decentralized network infrastructure is hence superior.

## Conclusion

This article considers solving the decentralized consensus optimization problem with the parallel version multi-block ADM in a networked multi-agent system. The traditional ADM can be used but it requires the introduction of a second block of auxiliary variables whereas our method takes advantages of the problem's nature of having multiple blocks of variables. We analyze the rate of convergence of our method applied to the average consensus problem. Analysis results that the two-block ADM is a special case of the multi-block ADM on average consensus. With extensive numerical experiments, we demonstrate the effectiveness of the proposed algorithm.

In the implementation of a networked multi-agent system, practical issues such as packet loss, asynchronization, and quantization are inevitable. This article assumes that the communication links are reliable, the network time is slotted and well synchronized, and the exchanged information is not quantized. We would like to address these issues in future research.

## Appendix 1

This section provides some theoretical results in the article.

### Development of MB-ADM

The decentralized consensus optimization problem (2) with neighboring consensus constraints can be rewritten as the form of (5). Apparently,  $g_i = f_i$ ,  $\theta_i = x^{(i)}$ , and  $e$  is an  $L^2N \times 1$  zero vector. Each  $D_i$  is an  $L^2N \times N$  matrix with  $L^2$  blocks of  $N \times N$  matrices. Each block of  $D_i$  can be defined as follows. Consider an  $L \times L$  matrix  $U^{(i)}$ , whose  $(i', j')$ th entry  $U_{i'j'}^{(i)} = 1$  if  $j' = i$  and  $i' \in \mathcal{N}_i$ ,  $U_{i'j'}^{(i)} = -1$  if  $i' = i$  and

$j' \in \mathcal{N}_i$ , and  $U_{i'j'}^{(i)} = 0$  otherwise. The  $(i' + Lj' - L)$ th block of  $D_i$  is  $U_{i'j'}^{(i)}I_N$ , where  $I_N$  is an  $N \times N$  identity matrix. Substituting them to the multi-block ADM, we can find that in optimizing  $x^{(i)}$ , agent  $i$  only needs its local information as well as part of  $q$ ; on the other hand, to update its corresponding part of  $q$  and  $\lambda$ , each agent only needs based on the information from itself and its neighbors. The resulting algorithm is hence fully decentralized due to the nice structure of  $\{D_i\}$ .

At time  $t$ , the multi-block ADM works as follows:

*Step 1: Updating the auxiliary variables  $\{q_{ij}\}$ :*

$$q_{ij}(t+1) = \lambda_{ij}(t) + \beta \left( x^{(i)}(t) - x^{(j)}(t) \right), \quad \forall i, \forall j \in \mathcal{N}_i. \quad (8)$$

*Step 2: Optimizing the local copies  $\{x^{(i)}\}$ :*

$$x^{(i)}(t+1) = \arg \min_{x^{(i)}} f_i(x^{(i)}) + \sum_{j \in \mathcal{N}_i} (q_{ij}(t+1) - q_{ji}(t+1))^T x^{(i)} + \mu |\mathcal{N}_i| \|x^{(i)} - x^{(i)}(t)\|_2^2, \quad \forall i. \quad (9)$$

*Step 3: Updating the Lagrange multipliers  $\{\lambda_{ij}\}$ :*

$$\lambda_{ij}(t+1) = \lambda_{ij}(t) + \beta \left( x^{(i)}(t+1) - x^{(j)}(t+1) \right), \quad \forall i, \forall j \in \mathcal{N}_i. \quad (10)$$

Note that  $\beta$  and  $\mu$  are positive constant parameters used by the multi-block ADM.

The updating rules (8), (9), and (10) can also be further simplified. Since we often set  $\{\lambda_{ij}(0)\}$  as 0, (8) and (10) imply that  $q_{ij}(t+1) = -q_{ji}(t+1)$  and  $\lambda_{ij}(t+1) = -\lambda_{ji}(t+1)$ . Summing up the two sides of (8) and (10) and defining a new auxiliary variable  $q_i = \sum_{j \in \mathcal{N}_i} q_{ij}$  as well as a new Lagrange multiplier  $\lambda_i = \sum_{j \in \mathcal{N}_i} \lambda_{ij}$ , their updating rules are:

$$q_i(t+1) = \lambda_i(t) + \beta |\mathcal{N}_i| x^{(i)}(t) - \beta \sum_{j \in \mathcal{N}_i} x^{(j)}(t), \quad (11)$$

$$\lambda_i(t+1) = \lambda_i(t) + \beta |\mathcal{N}_i| x^{(i)}(t+1) - \beta \sum_{j \in \mathcal{N}_i} x^{(j)}(t+1). \quad (12)$$

Hence (9) simplifies to:

$$x^{(i)}(t+1) = \arg \min_{x^{(i)}} f_i(x^{(i)}) + 2q_i^T(t+1)x^{(i)} + \mu |\mathcal{N}_i| \|x^{(i)} - x^{(i)}(t)\|_2^2. \quad (13)$$



### State transition equation of MB-ADM

Combining the updating rules of  $q_i(t+1)$  and  $q_i(t)$  in (11) and the updating rule of  $\lambda_i(t)$  in (12), we get:

$$q_i(t+1) - q_i(t) = 2\beta|\mathcal{N}_i|x^{(i)}(t) - 2\beta \sum_{j \in \mathcal{N}_i} x^{(j)}(t) - \beta|\mathcal{N}_i|x^{(i)}(t-1) + \beta \sum_{j \in \mathcal{N}_i} x^{(j)}(t-1). \quad (14)$$

Substituting  $f_i(x^{(i)}) = \frac{1}{2}\|x^{(i)} - b_i\|_2^2$  into (13), the optimality condition for  $x^{(i)}(t+1)$  is:

$$x^{(i)}(t+1) + 2q_i(t+1) + 2\mu|\mathcal{N}_i|x^{(i)}(t+1) - 2\mu|\mathcal{N}_i|x^{(i)}(t) - b_i = 0. \quad (15)$$

Combining the optimality conditions of  $x^{(i)}(t+1)$  and  $x^{(i)}(t)$  and (14) leads to:

$$x^{(i)}(t+1) = \frac{1 - 4\beta|\mathcal{N}_i| + 4\mu|\mathcal{N}_i|}{1 + 2\mu|\mathcal{N}_i|}x^{(i)}(t) + \frac{4\beta}{1 + 2\mu|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} x^{(j)}(t) + \frac{2\beta|\mathcal{N}_i| - 2\mu|\mathcal{N}_i|}{1 + 2\mu|\mathcal{N}_i|}x^{(i)}(t-1) - \frac{2\beta}{1 + 2\mu|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} x^{(j)}(t-1). \quad (16)$$

The initial state is  $s_M(1) = \left[ \frac{b_1}{1+2\mu|\mathcal{N}_1|}, \dots, \frac{b_L}{1+2\mu|\mathcal{N}_L|}, 0, \dots, 0 \right]$  when each agent  $i$  initializes  $q_i(0) = 0$ ,  $x^{(i)}(0) = 0$ , and  $\lambda_i(0) = 0$ .

### Proof of Property 1 in Proposition 1

It is straightforward to show that  $\rho_{M1} = 1$  is an eigenvalue of  $\Phi_M$ , as well as  $l_{M1}$  and  $r_{M1}$  are its corresponding left and right eigenvectors. Next we prove that  $\rho_{M1} = 1$  is with multiplicity 1 by contradiction. If  $\rho_{M1} = 1$  belongs to a larger Jordan block, there exists a vector  $[w^T, \bar{w}^T]^T$ , such that  $\Phi_M[w^T, \bar{w}^T]^T = [w^T, \bar{w}^T]^T + [1, \dots, 1, 1, \dots, 1]^T$ . Here  $w$  and  $\bar{w}$  are both  $L \times 1$  vectors (see [36], Fact 2). Observing the lower half of  $\Phi_M$ , apparently  $\bar{w} = w - [1, \dots, 1]^T$ . Suppose that  $w_k$  has the largest real part among all elements

of  $w$ . Picking up the  $k$ th row of  $\Phi_M[w^T, \bar{w}^T]^T = [w^T, \bar{w}^T]^T + [1, \dots, 1, 1, \dots, 1]^T$ , we have:

$$\left( \frac{1 - 4\beta|\mathcal{N}_k| + 4\mu|\mathcal{N}_k|}{1 + 2\mu|\mathcal{N}_k|}w_k + \frac{2\beta|\mathcal{N}_k| - 2\mu|\mathcal{N}_k|}{1 + 2\mu|\mathcal{N}_k|}(w_k - 1) \right) + \sum_{j \in \mathcal{N}_k} \left( \frac{4\beta}{1 + 2\mu|\mathcal{N}_k|}w_j - \frac{2\beta}{1 + 2\mu|\mathcal{N}_k|}(w_j - 1) \right) = w_k + 1, \quad (17)$$

or equivalent to:

$$\frac{2\mu|\mathcal{N}_k|}{1 + 2\mu|\mathcal{N}_k|} + \sum_{j \in \mathcal{N}_k} \frac{2\beta}{1 + 2\mu|\mathcal{N}_k|}w_j = 1 + \frac{2\beta|\mathcal{N}_k|}{1 + 2\mu|\mathcal{N}_k|}w_i. \quad (18)$$

Denote the real part of  $w_k$  and  $w_j$  as  $\text{Re}(w_k)$  and  $\text{Re}(w_j)$ , respectively. Recalling that  $\text{Re}(w_k) \geq \text{Re}(w_j)$  and picking up the real part of (18), we have:

$$1 + \frac{2\beta|\mathcal{N}_k|}{1 + 2\mu|\mathcal{N}_k|}\text{Re}(w_i) = \frac{2\mu|\mathcal{N}_k|}{1 + 2\mu|\mathcal{N}_k|} + \sum_{j \in \mathcal{N}_k} \frac{2\beta}{1 + 2\mu|\mathcal{N}_k|}\text{Re}(w_j) \leq \frac{2\mu|\mathcal{N}_k|}{1 + 2\mu|\mathcal{N}_k|} + \sum_{j \in \mathcal{N}_k} \frac{2\beta}{1 + 2\mu|\mathcal{N}_k|}\text{Re}(w_k) = \frac{2\mu|\mathcal{N}_k|}{1 + 2\mu|\mathcal{N}_k|} + \sum_{j \in \mathcal{N}_k} \frac{2\beta|\mathcal{N}_k|}{1 + 2\mu|\mathcal{N}_k|}. \quad (19)$$

This leads to contradiction. Hence  $\rho_{M1} = 1$  is an eigenvalue of  $\Phi_M$  with multiplicity 1.

### Proof of Proposition 2

Denote the  $i$ th eigenvalue of  $\Phi_i$  as  $\rho_{Mi}$ . Apparently, its eigenvectors should have the form of  $[\rho_{Mi}v^T, v^T]^T$  where  $v^T = [v_1, \dots, v_L]^T$  is a nonzero vector, since the lower half of  $\Phi_M$  is  $[I_{L \times L}, 0_{L \times L}]$ . Suppose that  $v_k$  has the largest norm (here we use  $|\cdot|$  to denote the norm of a complex number) among all elements of  $v$ . Then picking up the  $k$ th row of  $\Phi_M[\rho_{Mi}v^T, v^T]^T = \rho_{Mi}[\rho_{Mi}v^T, v^T]^T$ , we have:

$$\left( \frac{1 - 4\beta|\mathcal{N}_k| + 4\mu|\mathcal{N}_k|}{1 + 2\mu|\mathcal{N}_k|}\rho_{Mi} + \frac{2\beta|\mathcal{N}_k| - 2\mu|\mathcal{N}_k|}{1 + 2\mu|\mathcal{N}_k|} \right)v_k + \sum_{j \in \mathcal{N}_k} \left( \frac{4\beta}{1 + 2\mu|\mathcal{N}_k|}\rho_{Mi} - \frac{2\beta}{1 + 2\mu|\mathcal{N}_k|} \right)v_j = \rho_{Mi}^2 v_k, \quad (20)$$

or equivalently:

$$\begin{aligned} & \left( \rho_{Mi}^2 - \frac{1 - 4\beta|\mathcal{N}_k| + 4\mu|\mathcal{N}_k|}{1 + 2\mu|\mathcal{N}_k|} \rho_{Mi} - \frac{2\beta|\mathcal{N}_k| - 2\mu|\mathcal{N}_k|}{1 + 2\mu|\mathcal{N}_k|} \right) v_k \\ &= \sum_{j \in \mathcal{N}_k} \left( \frac{4\beta}{1 + 2\mu|\mathcal{N}_k|} \rho_{Mi} - \frac{2\beta}{1 + 2\mu|\mathcal{N}_k|} \right) v_j. \end{aligned} \quad (21)$$

Since  $v_k$  has the largest norm among all elements of  $v$ , taking norms for the both sides of (21) leads to:

$$\begin{aligned} & \left| \left( \rho_{Mi}^2 - \frac{1 - 4\beta|\mathcal{N}_k| + 4\mu|\mathcal{N}_k|}{1 + 2\mu|\mathcal{N}_k|} \rho_{Mi} - \frac{2\beta|\mathcal{N}_k| - 2\mu|\mathcal{N}_k|}{1 + 2\mu|\mathcal{N}_k|} \right) \right| |v_k| \\ &= \left| \sum_{j \in \mathcal{N}_k} \left( \frac{4\beta}{1 + 2\mu|\mathcal{N}_k|} \rho_{Mi} - \frac{2\beta}{1 + 2\mu|\mathcal{N}_k|} \right) v_j \right| \\ &\leq \sum_{j \in \mathcal{N}_k} \left| \left( \frac{4\beta}{1 + 2\mu|\mathcal{N}_k|} \rho_{Mi} - \frac{2\beta}{1 + 2\mu|\mathcal{N}_k|} \right) v_j \right| \\ &\leq \sum_{j \in \mathcal{N}_k} \left| \frac{4\beta}{1 + 2\mu|\mathcal{N}_k|} \rho_{Mi} - \frac{2\beta}{1 + 2\mu|\mathcal{N}_k|} \right| |v_j| \\ &\leq \left| \frac{4\beta|\mathcal{N}_k|}{1 + 2\mu|\mathcal{N}_k|} \rho_{Mi} - \frac{2\beta|\mathcal{N}_k|}{1 + 2\mu|\mathcal{N}_k|} \right| |v_k|. \end{aligned} \quad (22)$$

Notice that the inequalities turn to equalities when and only when  $v_j = v_k, \forall j \in \mathcal{N}_k$ . As  $v_k$  has the largest norm among all elements of  $v$ , any  $v_j$  with  $j \in \mathcal{N}_k$  also has such inequalities, and the inequalities turn to equalities when and only when  $v_{j'} = v_j, \forall j' \in \mathcal{N}_j$ . Because the network is connected, we can deduce that these inequalities turn to equalities when and only when  $\{v_i\}$  are all equal. This corresponds to the eigenvalue  $\rho_{M1} = 1$ . Canceling  $|v_k|$  from the both sides and defining  $d_1 = \frac{2\beta|\mathcal{N}_k|}{1+2\mu|\mathcal{N}_k|}$  and  $d_2 = \frac{2\mu|\mathcal{N}_k|}{1+2\mu|\mathcal{N}_k|}$ , (22) is equivalent to:

$$|\rho_{Mi}^2 - \rho_{Mi} + 2d_1\rho_{Mi} - d_2\rho_{Mi} - d_1 + d_2| \leq |2d_1\rho_{Mi} - d_1|. \quad (23)$$

Let us consider the two nontrivial special cases.

**Case 1.** The parameters  $\mu$  and  $\beta$  are chosen such that  $\mu = 2\beta > 0$ ; further,  $\frac{2\beta|\mathcal{N}_j|}{1+2\mu|\mathcal{N}_j|} < \frac{1}{4}$  and  $\frac{2\mu|\mathcal{N}_j|}{1+2\mu|\mathcal{N}_j|} < \frac{1}{2}$  for all  $j = 1, 2, \dots, L$ .

In this case,  $d_1 = \frac{2\beta|\mathcal{N}_k|}{1+2\mu|\mathcal{N}_k|} < \frac{1}{4}$  and  $d_2 = \frac{2\mu|\mathcal{N}_k|}{1+2\mu|\mathcal{N}_k|} < \frac{1}{2}$ . Let us choose  $d = d_1 = \frac{d_2}{2}, \frac{1}{4} > d > 0$ . Hence, (23) simplifies to:

$$|\rho_{Mi}^2 - \rho_{Mi} + d| \leq |2d\rho_{Mi} - d|. \quad (24)$$

Define  $w = \rho_{Mi} - \frac{1}{2}$ , we have:

$$\begin{aligned} & |w^2 + d - \frac{1}{4}| \leq 2d|w| \\ \Rightarrow & |w|^2 + d - \frac{1}{4} \leq 2d|w| \\ \Rightarrow & |w|^2 - 2d|w| + d^2 \leq d^2 - d + \frac{1}{4} \\ \Rightarrow & (|w| - d)^2 \leq (d - \frac{1}{2})^2 \\ \Rightarrow & |w| \leq \frac{1}{2} \end{aligned} \quad (25)$$

Recall that  $|w| = \frac{1}{2}$  only for  $\rho_{M1} = 1$ . For any other eigenvalues,  $|w| < \frac{1}{2}$ , and hence  $|\rho_{Mi}| < 1$  for  $i \neq 1$ .

**Case 2.** The parameters  $\mu$  and  $\beta$  are chosen such that  $\mu = \beta > 0$ ; further,  $\frac{2\beta|\mathcal{N}_j|}{1+2\mu|\mathcal{N}_j|} < \frac{1}{2}$  and  $\frac{2\mu|\mathcal{N}_j|}{1+2\mu|\mathcal{N}_j|} < \frac{1}{2}$  for all  $j = 1, 2, \dots, L$ .

In this case,  $d_1 = \frac{2\beta|\mathcal{N}_k|}{1+2\mu|\mathcal{N}_k|} < \frac{1}{2}$  and  $d_2 = \frac{2\mu|\mathcal{N}_k|}{1+2\mu|\mathcal{N}_k|} < \frac{1}{2}$ . Let us choose  $d = d_1 = d_2, \frac{1}{2} > d > 0$ . Hence, (23) simplifies to:

$$|\rho_{Mi}^2 - \rho_{Mi} + d\rho_{Mi}| \leq |2d\rho_{Mi} - d|. \quad (26)$$

Let us prove the conclusion by contradiction. Suppose that there exists a  $\rho_{Mi}$  with  $|\rho_{Mi}| \geq 1$  satisfies (26), then:

$$\begin{aligned} & |\rho_{Mi}^2 - \rho_{Mi} + d\rho_{Mi}| \leq |2d\rho_{Mi} - d| \\ \Rightarrow & |\rho_{Mi} - 1 + d| \leq |2d\rho_{Mi} - d| \\ \Rightarrow & |\rho_{Mi} - \frac{1}{2} - \frac{1}{2} + d| \leq 2d|\rho_{Mi} - \frac{1}{2}| \\ \Rightarrow & |\rho_{Mi} - \frac{1}{2}| \leq \frac{1}{2} \\ \Rightarrow & |\rho_{Mi}| \leq 1 \end{aligned} \quad (27)$$

Again, the inequalities turns to equalities only for  $\rho_{M1} = 1$ . For any other eigenvalue  $\rho_{Mi}$ , we have  $|\rho_{Mi}| < 1$  which contradicts with  $|\rho_{Mi}| \geq 1$ . Therefore,  $|\rho_{Mi}| < 1$  for  $i \neq 1$ .

#### State transition equation of TB-ADM

Substituting  $f_i(x^{(i)}) = \frac{1}{2} \|x^{(i)} - b_i\|_2^2$  into:

$$\begin{aligned} x^{(i)}(t+1) &= \arg \min_{x^{(i)}} f_i(x^{(i)}) + \alpha_i^T(t) x^{(i)} + c \sum_{j \in \mathcal{N}_i} \|x^{(i)} \\ &\quad - \frac{1}{2} (x^{(i)}(t) + x^{(j)}(t))\|_2^2, \end{aligned}$$

the optimality condition for  $x^{(i)}(t+1)$  is:

$$(1 + 2c|\mathcal{N}_i|)x^{(i)}(t+1) - c|\mathcal{N}_i|x^{(i)}(t) - \sum_{j \in \mathcal{N}_i} x^{(j)}(t) + \alpha_i(t) - b_i = 0. \quad (28)$$

Considering  $x^{(i)}(t)$ , the optimality condition is correspondingly:

$$(1 + 2c|\mathcal{N}_i|)x^{(i)}(t) - c|\mathcal{N}_i|x^{(i)}(t-1) - c \sum_{j \in \mathcal{N}_i} x^{(j)}(t-1) + \alpha_i(t-1) - b_i = 0. \quad (29)$$

Combining (28) and (29) with:

$$\alpha_i(t+1) = \alpha_i(t) + c|\mathcal{N}_i|x^{(i)}(t+1) - c \sum_{j \in \mathcal{N}_i} x^{(j)}(t+1),$$

the state transition equation for agent  $i$  is:

$$x^{(i)}(t+1) = x^{(i)}(t) + \frac{2c}{1 + 2c|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} x^{(j)}(t) - \frac{c|\mathcal{N}_i|}{1 + 2c|\mathcal{N}_i|} x^{(i)}(t-1) - \frac{c}{1 + 2c|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} x^{(j)}(t-1). \quad (30)$$

#### Competing interests

The authors declare that they have no competing interests.

#### Acknowledgements

The work of Qing Ling is supported in part by NSFC grant 61004137 and Fundamental Research Funds for the Central Universities. The work of Wotao Yin is supported in part by ARL and ARO grant W911NF-09-1-0383 and NSF grants DMS-0748839 and ECCS-1028790. The work of Xiaoming Yuan is supported in part by the General Research Fund No. 203311 from Hong Kong Research Grants Council.

#### Author details

<sup>1</sup>Department of Automation, University of Science and Technology of China, Hefei, Anhui, China. <sup>2</sup>School of Science, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu, China. <sup>3</sup>Department of Computational and Applied Mathematics, Rice University, Houston, Texas, USA. <sup>4</sup>Department of Mathematics, HongKong Baptist University, Kowloon Tong, Hong Kong.

Received: 15 February 2012 Accepted: 5 October 2012  
 Published: 12 November 2012

#### References

- W Ren, R Beard, E Atkins, Information consensus in multivehicle cooperative control: collective group behavior through local interaction. *IEEE Control Syst. Mag.* **27**, 71–82 (2007)
- R Olfati-Saber, in *Proceedings of CDC*. Kalman-consensus filter: optimality, stability, and performance, Shanghai, China, 2009, pp. 7036–7042
- S Boyd, N Parikh, E Chu, B Peleato, J Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundation Trends Mach. Learn.* **3**, 1–122 (2010)
- J Tsitsiklis, *Problems in decentralized decision making and computation*. (MIT, Ph.D Thesis, 1984)
- A Nedic, A Ozdaglar, Distributed subgradient methods for multi-agent optimization. *IEEE Trans. Autom. Control.* **54**, 48–61 (2009)
- K Srivastava, A Nedic, Distributed asynchronous constrained stochastic optimization. *IEEE J. Sel. Topics Signal Process.* **5**, 772–790 (2011)
- M Rabbat, R Nowak, in *Proceedings of IPSN*. Distributed optimization in sensor networks, Berkeley, USA, 2004, pp. 20–27
- M Rabbat, R Nowak, Quantized incremental algorithms for distributed optimization. *IEEE J. Sel. Areas Commun.* **23**, 798–808 (2006)
- L Xiao, S Boyd, S Kim, Distributed average consensus with least-mean-square deviation. *J. Parallel Distrib. Comput.* **67**, 33–46 (2007)
- S Kar, J Moura, Distributed consensus algorithms in sensor networks: quantized data and random link failures. *IEEE Trans. Signal Process.* **58**, 1383–1400 (2010)
- A Olshevsky, *Efficient Information Aggregation Strategies for Distributed Control and Signal Processing*. (Ph.D Thesis, MIT, 2010)
- I Schizas, A Ribeiro, G Giannakis, Consensus in ad hoc WSNs with noisy links - Part I: distributed estimation of deterministic signals. *IEEE Trans. Signal Process.* **56**, 350–364 (2008)
- G Mateos, I Schizas, G Giannakis, Distributed recursive least-squares for consensus-based in-network adaptive estimation. *IEEE Trans. Signal Process.* **57**, 4583–4588 (2009)
- Q Ling, Z Tian, Decentralized sparse signal recovery for compressive sleeping wireless sensor networks. *IEEE Trans. Signal Process.* **58**, 3816–3827 (2010)
- J Bazerque, G Giannakis, Distributed spectrum sensing for cognitive radio networks by exploiting sparsity. *IEEE Trans. Signal Process.* **58**, 1847–1862 (2010)
- G Mateos, J Bazerque, G Giannakis, Distributed sparse linear regression. *IEEE Trans. Signal Process.* **58**, 5262–5276 (2010)
- D Jakovetic, J Xavier, J Moura, Cooperative convex optimization in networked systems: augmented Lagrangian algorithms with direct gossip communication. *IEEE Trans. Signal Process.* **59**, 3889–3902 (2011)
- M Cetin, L Chen, J Fisher III, A Ihler, R Moss, M Wainwright, A Willsky, Distributed fusion in sensor networks. *IEEE Signal Process. Mag.* **23**, 42–55 (2006)
- J Predd, S Kulkarni, V Poor, Distributed learning in wireless sensor networks. *IEEE Signal Process. Mag.* **24**, 56–69 (2007)
- J Predd, S Kulkarni, H Poor, A collaborative training algorithm for distributed learning. *IEEE Trans. Inf. Theory.* **55**, 1856–1871 (2009)
- U Khan, S Kar, J Moura, Higher dimensional consensus: learning in large-scale networks. *IEEE Trans. Signal Process.* **58**, 2836–2849 (2010)
- A Jadbabaie, A Ozdaglar, M Zargham, in *Proceedings of CDC*. A distributed Newton method for network optimization, Shanghai, China, 2009, pp. 2736–2741
- J Koshal, A Nedic, U Shanbhag, Multiuser optimization: distributed algorithms and error analysis. *SIAM J. Optimiz.* **21**, 1046–1081 (2011)
- P Wan, M Lemmon, in *Proceedings of ACC*. Distributed network utility maximization using event-triggered augmented Lagrangian methods, St. Louis, USA, 2009, pp. 3298–3303
- F Zhao, J Shin, J Reich, Information-driven dynamic sensor collaboration. *IEEE Signal Process. Mag.* **19**, 61–72 (2002)
- F Zhao, L Guibas, *Wireless Sensor Networks: an Information Processing Approach*. (Morgan Kaufmann, Burlington, USA, 2004)
- I Schizas, G Giannakis, S Roumeliotis, A Ribeiro, Consensus in ad hoc WSNs with noisy links – part II: distributed estimation and smoothing of random signals. *IEEE Trans. Signal Process.* **56**(4), 1650–1666 (2008)
- A Ribeiro, I Schizas, S Roumeliotis, G Giannakis, Kalman filtering in wireless sensor networks: reducing communication cost in state estimation problems. *IEEE Control Syst. Mag.* **30**, 66–86 (2010)
- M Tao, Some parallel splitting methods for separable convex programming with  $O(1/t)$  convergence rate. in press
- B He, M Tao, M Xu, X Yuan, Alternating directions based contraction method for generally separable linearly constrained convex programming problems. in press
- B He, M Tao, X Yuan, Alternating direction method with Gaussian back substitution for separable convex programming. *SIAM J. Optim.* **22**, 313–340 (2012)
- D Bertsekas, J Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, 2nd edn. (Athena Scientific, Nashua, USA, 1997)
- D Bertsekas, *Numerical Optimization*, 2nd edn. (Athena Scientific, Nashua, USA, 1999)

34. B He, X Yuan, On the  $O(1/n)$  convergence rate of Douglas-Rachford alternating direction method. *SIAM J. Num. Anal.* **50**, 700–709 (2012)
35. T Erseghe, D Zennaro, E Dall'Anese, L Vangelista, Fast consensus by the alternating direction multipliers method. *IEEE Trans. Signal Process.* **59**, 5523–5537 (2011)
36. J Rosenthal, Convergence rates for Markov chains. *SIAM Rev.* **37**, 387–405 (1995)
37. S Boyd, A Ghosh, B Prabhakar, D Shah, Randomized gossip algorithms. *IEEE Trans. Inf. Theory.* **52**, 2508–2530 (2006)

doi:10.1186/1687-1499-2012-338

**Cite this article as:** Ling *et al.*: A multi-block alternating direction method with parallel splitting for decentralized consensus optimization. *EURASIP Journal on Wireless Communications and Networking* 2012 **2012**:338.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Immediate publication on acceptance
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

---

Submit your next manuscript at ▶ [springeropen.com](http://springeropen.com)

---