Forty-Eighth Annual Allerton Conference
Allerton House, UIUC, Illinois, USA
September 29 - October 1, 2010

Control Approach to Distributed Optimization

Jing Wang and Nicola Elia

Abstract— **In this paper, we propose a novel computation model for solving the distributed optimization problem where the objective function is formed by the sum of convex functions available to individual agent. Our approach differentiates from the existing approach by local convex mixing and gradient searching in that we force the states of the model to the global optimal point by controlling the subgradient of the global optimal function. In this way, the model we proposed does not suffer from the limitation of diminishing step size in gradient searching and allows fast asymptotic convergence. The model also shows robustness to additive noise, which is a main curse for algorithms based on convex mixing or consensus.**

Keywords: Distributed optimization, small gain theorem, Laplacian, additive noise, subgradients.

I. Introduction

The wide applications of peer to peer, sensor and ad hoc networks has necessitated the design and analysis of distributed algorithms that use local computation and information exchange to achieve global objectives. One particular interesting problem falling into this category is the distributed optimization problem where the objective function is formed by the sum of convex functions, each of which is *only* available to individual node. The objective of each node is to get the optimal solution of the global convex function via information exchange between their neighbors and local computation. This problem arises in a variety of applications such as distributed multi-agent coordination, estimation problems in sensor networks, recourse allocation problems and large-scale optimization problems in machine learning, see, e.g. [1], [2], [3], [4] and references therein.

To solve this unconstraint optimization problem, a computation model using the local subgradient and convex mixing of local information was first proposed in [1]. The analysis of this model incorporating randomness in network connectivity was provided in [2]. More recently, some extensions and modifications of this model are proposed to handle the problem over constraint sets, see, [3], [4], [5]. In this paper, in contrast to the methods proposed so far, we provide analysis and design of a novel computation model based on controlling the sum of the subgradients of individual convex functions. The model proposed in this paper is related to our previous work on mitigating the noise effect on distributed averaging [10], where in that paper, we show that additive noise to the consensus system can be mitigated by introducing two Laplacian matrices in the feedback loop. The difference is that besides using two Laplacian matrices in the loop, in this paper, we use extra control strategies for the sum of the subgradients, which drives the state of each node to the global optimal point. Our model overcomes the two limitations of the current algorithms, namely, the diminishing step size, which is a fundamental limitation of the performance of subgradient algorithms, and sensitivity to additive noise, which is an intrinsic property of consensus algorithms based on convex mixing.

Our work is also related to the seminal work of Tsitsiklis and his colleagues [6], [7], [8], where they analyzed algorithms for minimizing a smooth function while distribute the processing of the vector components to several agents. The problem of distribute the computation of a central optimization problem was also considered in [9] where an algorithm called message-passing algorithm was developed to tackle this problem. In comparison with these works, the problem we consider not aims to distribute the computation burden, but has an intrinsic need for collaboration among all agents due to the lack of central information of the objective function.

The paper is organized as follows. In section II, we define the problem considered in this paper. In section III, we analyze the continuous time model ans show its convergence property via LaSalle's invariance theorem. In section IV, we consider the corresponding discrete time model. The task of showing convergence in this case is more difficult than the continuous counterpart as we need to bound the induced gain of the subgradients. By appropriate design of certain scaling parameters, we show the states of each node reach the global optimal point through an application of the small gain theorem. We use simulation examples to demonstrate the two features of our model, namely fast convergence speed and resilience to noise, in section V. Finally, we conclude the paper and discuss some interesting extensions of the current work.

II. Problem Formulation

In this section, we provide a formal description of the distributed optimization problem and some key assumptions.

We consider a network consists of m nodes, labeled by the set $V = \{1, 2, \dots, m\}$. The topology of the network is represented by the undirected graph $\mathcal{G} = (V, E)$ with the edge set $E \subset V \times V$. The objective of each node is to find the solution of the following optimization problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{m} f_i(x) \\ \text{subject to} & x \in \mathbb{R}^n, \end{array} \tag{1}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$ is a convex function which is only available to node i. The following assumption is adopted throughout the paper.

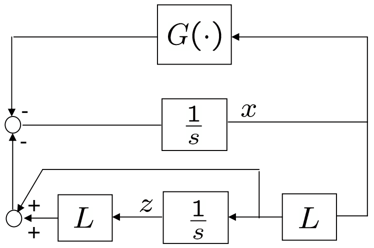


Fig. 1. Block diagram of the continuous time distributed computation model. For simplicity we consider $n = 1$ and the block diagram of more general cases can be obtained by incorporating Kronecker product. We view the subgradient $g_i(\cdot)$ as a map between $\mathbb{R}^n$ and $\mathbb{R}^n$. In this way, the map $G(\cdot)$ is diagonal whose elements are $g_i(\cdot)$.

Assumption 2.1: The optimal value of this problem, namely f^*, is finite and the optimal solution set

$$X^* = \{x \in \mathbb{R} \mid \sum_{i=1}^{m} f_i(x) = f^*\}$$

is nonempty and compact[1].

In this setting, since each node only has local access to f_i, they need to collaborate to solve this optimization problem. This involves local computation and information exchange with their neighbors, and the information flow pattern is determined by the graph $\mathcal{G}$. We define the neighbor set of node i as

$$N_i = \{j \mid (i,j) \in E\}.$$

We also put the following assumption on the graph $\mathcal{G}$.

Assumption 2.2: G is connected.
This assumption means that the information sent from each node will eventually be obtained by every other node through a directed path.

Problems (1) arises in various applications in the domain of information science and engineering. An immediate toy example will be the following distributed estimation problem. Consider a sensor network deployed to estimate certain objects, each sensor has a local measurement, denoted by x_i, and the objective is to obtain the median of all of them. This problem can be cast as minimize the summation of $|x_i - y|$ over the optimization variable y. For other applications in machine learning, see [4].

III. The Continuous Time Model

In this section, we develop the continuous time dynamic model to solve problem (1). Although this model can not be implemented on digital computers, it demonstrates the fundamental mechanism of our method: controlling the sum of subgradients to force the state to the optimal solution set. Besides, the proof of showing convergence is more elegant than the discrete time model which will be discussed later.

Let $x_i \in \mathbb{R}^n$ and $z_i \in \mathbb{R}^n$ be the state and auxiliary state of node i, we assume the dynamics of each node is governed by the following ordinary differential equations:

$$\begin{aligned}\dot{x}_i(t) &= \sum_{j \in N_i} a_{ij}(x_j(t) - x_i(t)) \\ &\quad + \sum_{j \in N_i} a_{ij}(z_j(t) - z_i(t)) - g_i(x_i(t)) \\ \dot{z}_i(t) &= \sum_{j \in N_i} a_{ij}(x_i(t) - x_j(t)), \end{aligned} \tag{2}$$

where $a_{ij} > 0$ are scalar weights chosen by node i, $g_i(t)$ is the subgradient of the function f_i at the point $x_i(t)$. We see from (2) that each node use its own states (x_i, z_i), the states of its neighbors and the subgradient of the local function f_i, which implies our model is distributed, i.e., every node only use local information.

The above equations for all nodes can be written compactly as

$$\begin{aligned}\dot{x}(t) &= -\mathbf{L}x(t) - \mathbf{L}z(t) - G(x(t)) \\ \dot{z}(t) &= \mathbf{L}x(t), \end{aligned} \tag{3}$$

where $\mathbf{L} = L \otimes I_n$ and $G(t)$ is a concatenation of $g_i(t)$. Here, L is the weighted graph Laplacian matrix(called Laplacian in short) and is defined as $L = [l_{ij}]$, where $l_{ii} = \sum_{j \in N_i} a_{ij}$ and $l_{ij} = a_{ij}$ for $j \in N_i$ and $l_{ij} = 0$ otherwise. We adopt the following assumption on the weights.

Assumption 3.1: The weights associated with (2) satisfy $a_{ij} = a_{ji}$ for all $i, j \in V$.

This assumption ensures that the Laplacian matrix L is symmetric. One simplest way to choose the weights is to set all positive weights to be 1. The Laplacian and its spectral properties are intimately related to the convergence property of the consensus algorithms (see, e.g., [12] for more detailed discussion). One could understand the overall system dynamics with the help of the block diagram of (3) shown in Figure 1.

In general, the subgradient $g_i(\cdot)$ is a nonlinear function of x_i, therefore, (3) could be seen as a nonlinear system. Let $\dot{x} = \dot{z} = 0$, the equilibrium points of (3) satisfy

$$x^* \in \text{span}\{\mathbf{1} \otimes v\} \quad \mathbf{L}z^* = -G(x^*), \tag{4}$$

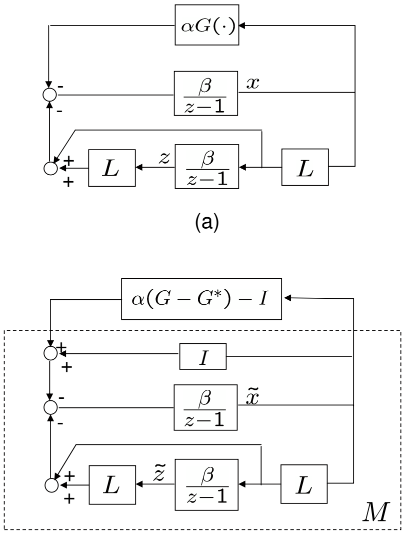
and

$$(\mathbf{1} \otimes v)' G(x^*) = v' \sum_{i=1}^{m} g_i(\hat{x}^*) = 0, \tag{5}$$

where v is an arbitrary vector in $\mathbb{R}^n$, $\mathbf{1} \in \mathbb{R}^m$ is a vector whose components are all ones and $x^* = \hat{x}^* \otimes \mathbf{1}$. (4) follows from $\mathbf{L}(\mathbf{1} \otimes v) = (L\mathbf{1}) \otimes (I_n v) = 0$[2], where we use the property that the row sums of L are all equal to zero. To get (5), we can multiply $(\mathbf{1} \otimes v)'$ on both sides of the state equaiton of x in (3). Since L is symmetric, $\mathbf{1}'L = 0$, and we have $(\mathbf{1} \otimes v)'\mathbf{L} = (\mathbf{1}'L) \otimes (v' I_n) = 0$. Since v is an arbitrary vector in $\mathbb{R}^n$, (5) implies that $\sum_{i=1}^{m} g_i(\hat{x}^*) = 0$,

[1] The compactness of the optimal solution set is a technical assumption to use LaSalle's invariance theorem in the future convergence analysis.

[2] Here, we use the property of Kroneker product that for matrices A, B, C, D with appropriate sizes, $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$, see, e.g., [13], chap. 13.



(a)

(b)

Fig. 2. Block diagram of the discrete time distributed computation model and its alternative transformation. Here, the transformation allows us to use small gain theorem to assess the convergence property of the system, and α and β are parameters to be designed to ensure the convergence.

i.e., for every equilibrium point x^*, each of its components $\hat{x}^*$ is the optimal solution for $f = \sum_{i=1}^{m} f_i$.

From Assumption 2.1, we can pick an arbitrary $\hat{x}^* \in X^*$ and let $x^* = \mathbf{1} \otimes \hat{x}^*$, then choose an z^* such that $Lz^* = -G(x^*)$, where the subgradients of each f_i at $\hat{x}^*$ are chosen such that $\sum_{i=1}^{m} g_i(\hat{x}^*) = 0$. In this way, (x^*, z^*) is an equilibrium point of (3). We are ready to state the first main result in this paper, which shows that under (2), each state $x_i(t)$ converges to the optimal solution set asymptotically.

Theorem 3.2: Let Assumption 2.1, 2.2 and 3.1 hold. For system (2), we have

$$\lim_{t\to\infty} x(t) = \mathbf{1} \otimes \hat{x},$$

where $\hat{x} \in X^*$

Remark 3.3: The major difference between our computation model and the one in [1] is that for any optimal solution $\hat{x}^*$, $x^* = \mathbf{1} \otimes \hat{x}^*$ is an equilibrium point of our system. This feature allows us to use control theories to establish the asymptotic convergence property of the model while previous methods only establish the convergence of local averaged states (see [4] and references therein). Besides, it is intuitive that when we discretize (3), it does not suffice from the limitation of diminishing step used in the literature, see, *e.g.,*[1], [4]. This will be verified in the next section.

IV. The Discrete Time Model

In this section, we focus on design and analysis of the discrete time version of (3), which make it possible to implement the algorithm on digital communication systems.

Our approach is viewing the subgradient $g_i(\cdot)$ as an input-output map and using small gain theorem to guarantee the convergence property of the system. The motivation of bounding the induced gain of g_i is roughly stated as the following. Suppose each g_i is an affine function of state x_i (when each f_i is a quadratic function), then the overall system is a Linear Time Invariant (LTI) system. As seen from Figure 1, the transfer matrix seen by G has one pole at the origin (one can verify this easily from the block diagram). Therefore, when we discretize this model, it will has a pole at 1. If the induced gain of G is large, then from the root locus argument, some poles of the overall system will migrate out of the unit disk and thus introduce instability to the system.

To solve problem (1), we propose the following iterative algorithm. For each node i, its state $x(k)$ and auxiliary state $z(k)$ are iterated according to

$$\begin{aligned} x_i(k+1) &= x_i(k) + \beta \sum_{j\in N_i} a_{ij}(x_j(k) - x_i(k)) \\ &\quad + \beta \sum_{j\in N_i} a_{ij}(z_j(t) - z_i(t)) - \beta\alpha g_i(x_i(k)) \\ z_i(k+1) &= z_i(k) + \beta \sum_{j\in N_i} a_{ij}(x_i(k) - x_j(k)), \qquad (6) \end{aligned}$$

where $k \geq 0$ are nonnegative integers, and $\beta > 0$ and $\alpha > 0$ are parameters to be designed to ensure the stability property of the system. Note that here, we use $\beta\alpha$ as the constant scaling of the local subgradients. The state equations for all nodes can be written as

$$\begin{aligned} x(k+1) &= x(k) - \beta\mathbf{L}x(k) - \beta\mathbf{L}z(k) - \beta\alpha G(x(k)) \\ z(k+1) &= z(k) + \beta\mathbf{L}x(k), \qquad (7) \end{aligned}$$

The block diagram of system (7) is shown in Figure 2(a), where two Laplacians are present in the feedback loop. This implies that during each time interval, there are two stages of information exchange and computation. In the first stage, the nodes exchanges the states x_i and compute the auxiliary states z_i; during the second stage, the nodes exchange the auxiliary states z_i and update the states x_i.

Let $x(k+1) = x(k)$ and $z(k+1) = z(k)$, the equilibrium point of the system (7) satisfies (4) and (5). Let $\tilde{x} = x - x^*$, $\tilde{z} = z - z^*$, we get

$$\begin{aligned} \tilde{x}(k+1) &= \tilde{x}(k) - \beta\mathbf{L}\tilde{x}(k) - \beta\mathbf{L}\tilde{z}(k) \\ &\quad - \beta(\alpha(G(x(k)) - G(x^*))) \\ \tilde{z}(k+1) &= \tilde{z}(k) + \beta\mathbf{L}x(k), \end{aligned}$$

The above system can be transformed as the interconnection between the operator $I - \alpha(G(\cdot) - G(x^*))$ and the system

$$M: \quad \begin{aligned} \tilde{x}(k+1) &= \tilde{x}(k) - \beta\mathbf{L}\tilde{x}(k) - \beta\mathbf{L}\tilde{z}(k) - \beta u(k) \\ \tilde{z}(k+1) &= \tilde{z}(k) + \beta\mathbf{L}x(k) \\ y(k) &= x(k), \end{aligned}$$

where we assume the system has u as input and y as output. This transformation is shown in Figure 2(b). The objective of this transformation is that when the system seen by $\alpha G(\cdot)$ (see Figure 2(a)) is in feedback with $-I$, it becomes stable for certain range of β (this system originally contains a pole at 1 as discussed before) and allows us to use small gain theorem to establish the convergence result.

We first provide a Lemma which shows M is bounded input bounded output (BIBO) stable by choosing β appropriately. Let $d_m = \max_i \sum_{j\in N_i} a_{ij}$ denote the maximum degree of each node, we have the following result.

Lemma 4.1: If $0 < \beta < \frac{2}{2d_m+1}$, M is BIBO stable.

This result shows that BIBO stability of the system seen by the operator $I - \alpha(G - G^*)$ can be guaranteed by choosing β appropriately. In the simple scenario when we set all weights to 1 and the network topology is spatially invariant, β can be chosen locally. Since $I - \alpha(G - G^*)$ is diagonal, we can define its 2-induced gain as

$$\gamma = \sup_{\hat{x}\in\mathbb{R}^n} \max_{i\in V} \frac{||\hat{x} - \hat{x}^* - \alpha(g_i(\hat{x}) - g_i(\hat{x}^*))||_2}{||\hat{x} - \hat{x}^*||_2}.$$

The following result establishes the convergence property of the model.

Theorem 4.2: Suppose $0 < \beta < \frac{2}{2d_m+1}$, if $||M||_{\mathcal{H}_\infty} \cdot \gamma < 1$, we have

$$\lim_{k\to\infty} x(k) = \mathbf{1} \otimes \hat{x},$$

where $\hat{x} \in X^*$.

Theorem 4.2 provides a centralized method for the design of the network, as the information of g_i and $||M||_{\mathcal{H}_\infty}$ is required. However in many simulation examples, we see $||M||_{\mathcal{H}_\infty} = 1$ for β in certain range. Besides, we may choose small α to bound the 2-induced gain of $I - \alpha(G - G^*)$, which is plausible at least for the case when each f_i is a quadratic function.

Since we use the model of [10] as a building block of the model proposed in this paper, our intuition is that our model is also shows resilient property to additive noise, as least for the case when the system is linear. This will be illustrated by the simulation example shown in the next section. We will explore this noise resilient property in our future work.

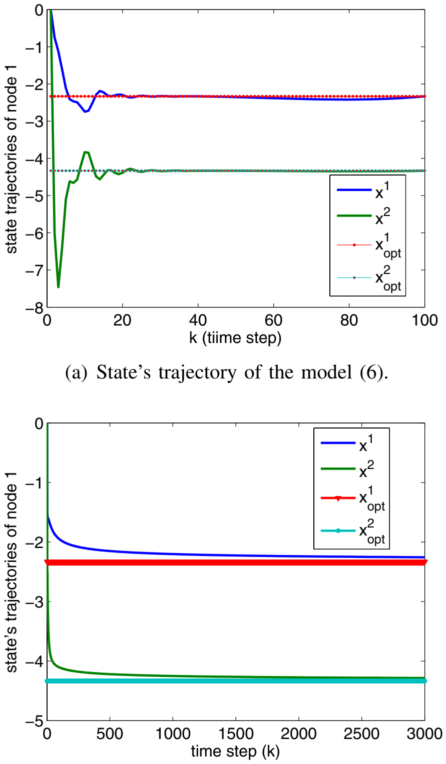
V. Simulations

In this section, we use an simulation example to show the two features of our model, namely, fast convergence speed and robustness to additive noise. We consider a ring graph containing 4 nodes. The private convex functions of each node is given by the quadratic function: $f_i(x) = x'P_ix + b_ix + c_i$, where

$$P_1 = \begin{bmatrix} 0.2 & 0.1 \\ 0.1 & 0.2 \end{bmatrix} \quad P_2 = \begin{bmatrix} 0.4 & 0.1 \\ 0.2 & 0.4 \end{bmatrix}$$

$$P_3 = \begin{bmatrix} 0.3 & 0.1 \\ 0.1 & 0.2 \end{bmatrix} \quad P_4 = \begin{bmatrix} 0.5 & 0.1 \\ 0.1 & 0.2 \end{bmatrix},$$

$b_1 = [18]'$, $b_1 = [11]'$, $b_1 = [31]'$ and $[51]'$ and c_i are chosen uniformly from $[0, 1]$. The optimal value is then given by $-(\sum_i P_i)^{-1}(\sum_i b_i) = [-2.3333 \; -4.3333]'$. We choose the model parameters as $\beta = 0.2$, $a_{ij} = 1$ for all $i \in V$ and $j \in N_i$, $\alpha = 3$. We compare the performance of our model with the model proposed in [1], where the double stochastic matrix defining the model is chosen to be $(I_4 - \beta L) \otimes I_2$ and the step size is chosen to be $1/\sqrt{k}$ (cf. [4]).



(a) State's trajectory of the model (6).

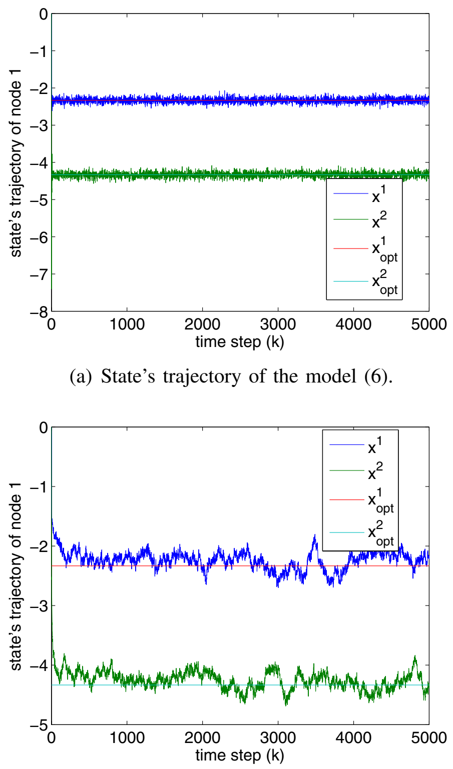
(b) State's trajectory of the model in [1].

Fig. 3. An example to compare the convergence speed between our model and the model proposed in [1]. It is shown that using our model, the states converge to the true optimal value in less than 30 iterations while the model of [1] requires more than 3000 iterations. Here, we only plot the state's trajectory of node 1, the states' trajectories of other nodes behave similarly.

The simulation results in Figure 3 show a remarkable improvement of the convergence speed of our model with respect to the gradient decent model. This is not surprising as our model does not need the diminishing step size. Our conjecture is that in certain cases, the convergence speed of our model could approximate that of the distributed consensus algorithm or mixing time of a Markov chain. Figure 4 illustrates the robustness of our model to additive noise, while the gradient model suffers from the random walk behavior. This suggest our model could also be used to compute the global optimal value even in the presence of noise by averaging the local states sequences, which will of practical value for the implementation when random disturbances is present.

VI. Conclusions and Discussion

In this paper, we have proposed a distributed computation model for optimize the global objective function using dynamic iteration and local information exchange. The model has fast convergence speed and robustness with respect to additive noise. There could be many extensions of this preliminary work. In this paper, we use the small gain theorem to bound the induced gain of the subgradients to show stability. This introduce some conservatism which



(a) State's trajectory of the model (6).

(b) State's trajectory of the model in [1].

Fig. 4. An example to compare the effect of additive noise on our model and the model of [1]. It is shown that for our model, the states converge to the true optimal value with bounded variance while the states of the subgradient decent model behaves as a random walk. The noise are white Gaussian with variance being 0.04.

may be removed by other approaches, for example, the scaling techniques in robust control to handle the μ problem. In this paper, we only use β as the design parameter to ensure stability. However, there could be systematic approach to design the system dynamics to ensure convergence, as introduced in [11], which shows that robust control method can be used to design agents for consensus. Other extensions include analyzing the convergence speed of the model and incorporating constraints to the optimization problem, which we plan to investigate in the future.

VII. Acknowledgment

This research has been supported by NSF under grant number ECS-0901846. The authors are with the Dept. of Electrical and Computer engineering, Iowa State University, Ames, IA 50010. Emails : {criswang, nelia}@iastate.edu.

References

[1] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Automat. Contr.*, vol. 54, Jan. 2009.

[2] I. Lobel and A. Ozdaglar, "Distributed subgradient methods over random networks," Tech. Report 2800, MIT LIDS, 2008.

[3] I. Lobel, A. Ozdaglar, and D. Feijer, "Distributed Multi-agent Optimization with State-Dependent Communication," LIDS report 2834, submitted for publication, 2010.

[4] J. C. Duchi, A. Agarwal and M. J. Weinwright, "Dual Averaging for Distributed Optimization: Convergence Analysis and Network Scaling," available on line: `http://arxiv.org/abs/1005.2012`.

[5] S.S. Ram, A. Nedic, and V. V. Veeravalli "Distributed Stochastic Subgradient Projection Algorithms for Convex Optimization," to appear in Journal of Optimization Theory and Applications, 2010.

[6] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, " Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Automat. Contr.*, vol. 31, pp. 803-812, Sep. 1986.

[7] J. N. Tsitsiklis, *Problems in decentralized decision making and computation*, Ph.D. Thesis, Massachusetts Institute of Technology, 1984.

[8] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall, Inc., Upper Saddle River, N.J., USA, 1989.

[9] C. C. Moallemi and B. Van Roy, "Convergence of min-sum message-passing for convex optimization," to appear in *IEEE Transaction on Information Theory*.

[10] J. Wang and N. Elia, "Distributed agreement in the presence of noise," in *Proceedings of the 47th annual Allerton conference on Communication, control, and computing*, Monticello, Illinois, USA, 2009, pp. 1575-1581.

[11] J. Wang and N. Elia, "Agents design for distributed consensus over Networks of Fixed and switching topologies," in *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, Shanghai, P.R. China, Dec. 16-18, 2009, pp. 5815-5820.

[12] R. Olfati-Saber and R. M. Murray, "Consensus problems in network of agents with switching topology and time-delays," *IEEE Trans. Automat. Contr.*, vol. 49, pp. 1520-1533, Sep. 2004.

[13] A. J. Laub, *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadephia, PA, 2005.

[14] H. K. Khalil, *Nonlinear Systems*, Prentice Hall, Englewood Cliffs, N.J., second edition, 1995.

[15] R. A. Horn and C.R. Johnson, *Matrix Analysis*, Cambridge University Press, 1987.