2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Scheduling for Finite Time Consensus

Chih-Kai Ko
Department of Electrical Engineering
California Institute of Technology
Pasadena, California 91125
Email: cko@caltech.edu

Ling Shi
Department of Electronic and Computer Engineering
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
Email: eesling@ust.hk

Abstract— **We study the problem of link scheduling for discrete-time agents to achieve average consensus in finite time under communication constraints. We provide necessary and sufficient conditions under which finite time consensus is possible. Furthermore, we prove bounds on the consensus time and exhibit provably optimal communication policies. We also discuss the dual problem of designing communication schedules given a fixed consensus-time requirement.**

I. Introduction

Multi-agent systems have attracted much attention in the past few years [1], [2]. The general research focus in this area is designing decentralized control laws to achieve certain global objective. For example, in classical consensus problems [3], groups of agents try to agree upon certain quantities such as their positions, environment temperature, etc., through the exchange of data with their neighbors. In multi-vehicle formation control problems [4], a group of vehicles try to main a certain desired formation by communicating with neighboring vehicles. Other examples include behavior of swarms [5], sensor network data fusion [6], unmanned aerial vehicles (UAVs), attitude alignment of satellite clusters, etc. [7], [8].

In this paper we consider the finite time average consensus problem. That is, we try to find efficient algorithms such that a collection of n agents reach consensus on the average of their initial values in finite time. Although our main focus is on the consensus time, we note that there are many other figures of merit for evaluating consensus algorithms. For example, robustness to imperfect communication and random distances. Issues such as communication delays and changes in the communication topology over time have been examined by Olfati-Saber et al. [9] and Ren et al. [10].

Rate of convergence is widely used as a measure of performance [11], [12] in consensus problems. Many tools that are used to derive and analyze the performance of consensus algorithms come from graph theory [13]. Graphs (possibly time-varying) provide an efficient representation of the communication topology between the group of agents, with each node representing an individual agent, and an edge representing the information link between a pair of agents. Distributed average consensus algorithms [3] in general have infinite time to reach consensus and its rate of convergence is given by the second smallest eigenvalue, λ_2, of the corresponding graph Laplacian. By optimally choosing how much weight each node should set for neighboring nodes' values, Yang et al. [14] sped up the rate of convergence, while trading-off robustness. Xiao et al. [15] provided similar results using convex optimization techniques but in a discrete-time setting. Olfati-Saber et al. [16] introduced additional communication links into the network and created small world networks to speed up the rate of convergence.

When finite-time consensus is possible, time to reach consensus is used as a measure of performance. Cortes [17] studied the application of non-smooth gradient flows for finite time consensus. Wang and Xiao [18] considered a finite-time state consensus problem for continuous-time multi-agent systems and provided two protocols for those agents reaching consensus. By employing finite-time Lyapunov functions, they derived conditions which guarantee that consensus is reached in finite-time when the two protocols are used. They also provided upper bounds on the time to reach consensus. Sundaram and Hadjicostis [19], [20] considered the discrete-time consensus problem and presented a method for each node achieving consensus in a finite number of times by linearly combining its own past values. Their approach requires that each node has sufficient computation capability. In many cases, computing the the optimal weighting matrix is computationally intractable.

Unlike most of the aforementioned results and approaches, the consensus algorithms proposed in this paper consisting of optimally designing the communication schedule between agents. That is, we specify the time varying communication topology at each instance in time. The main contributions of this paper are summarized as follows.

- We provide necessary and sufficient conditions for finite-time consensus.
- We analyze the scenarios of finite-time consensus under communication constraints, and provide bounds on the time to reach consensus.
- We provide several consensus algorithms, including ones that are provably optimal, i.e., they achieve the derived lowerbounds on the time to reach consensus.

The rest of the paper is organized as follows. In Section II, we provide the precise mathematical description of our problem. Then in Section III, we consider the scenario that only one pair of agents is allowed communicate at any given time. We provide a lower bound on the time to reach consensus and

978-1-4244-4524-0/09/$25.00 ©2009 AACC

an algorithm that achieves this lower bound. In Section IV, we generalize the results to many communicating agents. Again, we provide a lower bound on the time to reach consensus as well as algorithms that achieve the lower bound. Furthermore, we give necessary and sufficient conditions for the general finite-time consensus problems. Finally, in Section V, we consider the dual problem: what are the communication requirements to achieve consensus under a given time constraint?

II. Problem Statement

Consider a collection of n nodes: $V = \{0, 1, \ldots, n-1\}$. Let $x_i(t)$ denote the value of node i at time t. For conciseness, we define $\mathbf{x}(t) = [x_0(t), \ldots, x_{n-1}(t)]^{\mathsf{T}} \in \mathbb{R}^n$.

At each discrete time step t, the communication topology is specified by an undirected graph $G(t) = (V, E(t))$ where each edge $e \in E(t)$ denotes a communication link between two nodes and we allow for multiple edges.

For each node $i \in V$, define $E_i(t)$ as the set of edges incident on node i. The node values evolve according to:

$$x_i(t+1) = \frac{x_i(t)}{|E_i(t)|+1} + \sum_{j\,:\,(i,j)\in E_i(t)} \frac{x_j(t)}{|E_j(t)|+1} \qquad (1)$$

where $|\cdot|$ denotes the cardinality of a set.

Define a *communication schedule* $S = \{E(0), E(1), \ldots\}$ as a sequence of edge sets. Given a schedule S, its consensus time is

$$t_c(S) = \max_{\mathbf{x}(0)\in\mathbb{R}^n} \min \left\{ t : x_i(t) = \frac{1}{n}\sum_{i=0}^{n-1} x_i(0) \right\}$$

Clearly, not all schedules have finite consensus times.

If for all $E(t) \in S$, we have $|E(t)| \le k$ then we call S a *k-edge schedule*. Define $\mathcal{S}_k$ as the set of all k-edge schedules. Since it's often costly to establish communication links, we view k as a communication constraint and define

$$t_c(\mathcal{S}_k) = \min_{S\in\mathcal{S}_k} t_c(S)$$

and investigate the following:

- What are the conditions on n and k for a finite $t_c(\mathcal{S}_k)$?
- When it is finite, how does the consensus $t_c(\mathcal{S}_k)$ time vary with k?
- Can we characterize efficient consensus schedules?

III. Single-Edge Schedules

We begin with a study of consensus times for single edge schedules $\mathcal{S}_1$ when n is a power of 2.

Using a potential function argument, we can show that $t_c(\mathcal{S}_1) \ge (n \log n)/2$. But first, we require a brief information theory interlude.

A. Preliminary: Change of Entropy by Averaging

Let $\mathbf{p} = (p_1, p_2, \ldots, p_n)$ and $\mathbf{q} = (q_1, q_2, \ldots, q_n)$ be n-dimensional probability vectors. Let $H(\mathbf{p}) = -\sum_i p_i \log p_i$ denote the binary entropy function. Unless otherwise specified, all log's are base-2 and we adopt the convention that $0 \log 0 = 0$.

Because $H(\cdot)$ is concave (see Theorem 2.7.3 in [21]),

$$H\left(\frac{\mathbf{p}+\mathbf{q}}{2}\right) \ge \frac{1}{2} H(\mathbf{p}) + \frac{1}{2} H(\mathbf{q})$$

by Jensen's Inequality. So if we replaced both $\mathbf{p}$ and $\mathbf{q}$ by their average, the total entropy does not decrease:

$$\Delta H \equiv 2\, H\left(\frac{\mathbf{p}+\mathbf{q}}{2}\right) - (H(\mathbf{p}) + H(\mathbf{q})) \ge 0.$$

Let $D(\mathbf{p}\|\mathbf{q}) = \sum_i p_i \log(p_i/q_i)$ denote the Kullback-Leibler divergence between $\mathbf{p}$ and $\mathbf{q}$. We have

$$\begin{aligned} D(\mathbf{p}\|(\mathbf{p}+\mathbf{q})/2) &= \sum_i p_i \log \frac{p_i}{(p_i+q_i)/2} \\ &= \sum_i p_i + \sum_i p_i \log \frac{p_i}{p_i+q_i} \\ &\le 1 \end{aligned}$$

where the last inequality is because $p_i \le p_i + q_i$ so $\log(p_i/(p_i+q_i)) \le 0$. Now, we can upperbound the increase in entropy due to averaging:

$$\begin{aligned} \Delta H &= 2\, H\left(\frac{\mathbf{p}+\mathbf{q}}{2}\right) - (H(\mathbf{p}) + H(\mathbf{q})) \\ &= -2 \sum_i \left(\frac{p_i+q_i}{2} \log \frac{p_i+q_i}{2} \right) \\ &\quad + \sum_i p_i \log p_i + \sum_i q_i \log q_i \\ &= \sum_i \Big[-(p_i+q_i) \log \frac{p_i+q_i}{2} \\ &\quad + p_i \log p_i + q_i \log q_i \Big] \\ &= \sum_i \Big[p_i \left(\log p_i - \log \frac{p_i+q_i}{2} \right) \\ &\quad + q_i \left(\log q_i - \log \frac{p_i+q_i}{2} \right) \Big] \\ &= \sum_i p_i \log \frac{p_i}{(p_i+q_i)/2} + q_i \log \frac{q_i}{(p_i+q_i)/2} \\ &= D\left(\mathbf{p} \,\middle\|\, \frac{\mathbf{p}+\mathbf{q}}{2}\right) + D\left(\mathbf{q} \,\middle\|\, \frac{\mathbf{p}+\mathbf{q}}{2}\right) \\ &\le 2 \end{aligned}$$

Lemma 1. *The change in total entropy, ΔH, due to the averaging of two probability vectors is bounded by*

$$0 \le \Delta H \le 2$$

We remark that

$$\Delta H = 2 \cdot \mathrm{JS}(\mathbf{p}, \mathbf{q}) = \mathrm{D}\left(\mathbf{p} \,\middle\|\, \frac{\mathbf{p}+\mathbf{q}}{2}\right) + \mathrm{D}\left(\mathbf{q} \,\middle\|\, \frac{\mathbf{p}+\mathbf{q}}{2}\right)$$

where $\mathrm{JS}(\mathbf{p}, \mathbf{q})$ is the Jensen-Shannon divergence, a symmetrized version of the Kullbeck-Leibler divergence.

B. Lowerbound

Recall that $\mathbf{x}(t) = [x_0(t)\, x_1(t) \cdots x_{n-1}(t)]^\mathsf{T} \in \mathbb{R}^n$ denotes the node values at time t. For each node i, we can express its value at time t as

$$x_i(t) = \mathbf{p}_i(t)^\mathsf{T}\mathbf{x}(0)$$

where $\mathbf{p}_i(t)$ is a n-dimensional probability vector. Intuitively, $\mathbf{p}_i(t)$ represents the weighted contributions of $\mathbf{x}(0)$. Initially, for all i,

$$x_i(0) = \mathbf{p}_i(0)^\mathsf{T}\mathbf{x}(0) = \mathbf{e}_i^\mathsf{T}\mathbf{x}(0)$$

where $\mathbf{e}_i$ is the i-th column of the $n \times n$ identity matrix. When consensus is reached at some time, say t_c,

$$x_i(t_c) = \mathbf{p}_i(t_c)^\mathsf{T}\mathbf{x}(0) = \frac{1}{n}\mathbf{1}^\mathsf{T}\mathbf{x}(0)$$

for all i. Define $\phi_i(t) \triangleq H(\mathbf{p}_i(t))$ and $\phi(t) \triangleq \sum_{i=1}^n \phi_i(t)$ so that

$$\phi(t_c) = n\, H\left(n^{-1}\mathbf{1}\right) = n \log n.$$

Note that $\phi_i(0) = 0$. Since each averaging operation increases the total entropy by at most 2 (Lemma 1), we need at least $(n \log n)/2$ such operations to reach $\phi(t_c)$.

Theorem 2.

$$t_c(\mathcal{S}_1) \geq \frac{n \log n}{2}.$$

Since k vertex disjoint edges can increase entropy by at most $2k$, we can generalize this lowerbound:

Theorem 3. *If the k edges are vertex disjoint, then*

$$t_c(\mathcal{S}_k) \geq \frac{n \log n}{2k}$$

C. Upperbound

Consider the following scheduling algorithm for single edge consensus when the number of nodes is a power of 2.

```
Algorithm 1: SINGLEEDGECONSENSUS
  Input: $\{x_1, x_2, \ldots, x_n\}$
  Output: For all $i$, $x_i = n^{-1}\sum_{j=1}^n x_j$

for $i = 0$ to $\log n - 1$ do
    foreach $a, b \in \{0, 1, \ldots, n-1\}$ such that $a \oplus b = 2^i$ do
        $M = (x_a + x_b)/2$
        $x_a = M$
        $x_b = M$
    end
end
```

The "$\oplus$" in the algorithm denotes bit-wise XOR. The overall runtime of this strategy is $(n \log n)/2$ as the outer **for** loop executes $\log n$ times and the inner loop executes $n/2$ times. This gives us an upperbound on the required consensus time:

Theorem 4. *When n is a power of 2,*

$$t_c(\mathcal{S}_1) \leq \frac{n \log n}{2}.$$

If k is also a power of 2 and $k \leq n/2$, we can generalize this result to k vertex-disjoint edges at a time:

Theorem 5. *When n is a power of 2 and k is a power of 2,*

$$t_c(\mathcal{S}_k) \leq \frac{n \log n}{2k}.$$

Proof. The factor of k speedup occurs in the inner loop of Algorithm 1. Instead of performing consensus on 1 pair of vertices at a time, we can average k pairs. So the inner loop now requires $n/2k$ steps while the outer loop remains the same. □

The correctness of Algorithm 1 follows by recognizing that it is essentially a recursive algorithm which divides a size n problem into 2 problems of size $n/2$.

If k is not a power of 2 but $k \leq n/2$. We can omit some edges to get a runtime of $(n \log n)/2^{m_l+1}$ where $m_l = \lfloor \log_2 k \rfloor$. In the inner loop, we schedule only 2^{m_l} edges to get:

Theorem 6. *When n is a power of 2 and $k \leq n/2$*

$$\frac{n \log n}{2^{-m_u+1}} \leq t_c(\mathcal{S}_k) \leq \frac{n \log n}{2^{-m_l+1}}$$

where $m_l = \lfloor \log_2 k \rfloor$ and $m_u = \lceil \log_2 k \rceil$. Notice that when k is a power of 2, the bounds are tight.

Proof. Since $2^{m_u} \geq k$, we can lowerbound $t_c(\mathcal{S}_k)$ for k edges by the consensus time for 2^{m_u} edges. □

D. General n

To understand the behavior when n is not a power of 2, let's first examine the case where n is prime.

Lemma 7. *If n is a prime greater than 2, then one cannot achieve finite time consensus with $\mathcal{S}_1$.*

Proof. By contradiction, suppose that finite time consensus is possible. Consider initial node values:

$$x_i(0) = \begin{cases} n & \text{if } i = 0 \\ 0 & \text{otherwise} \end{cases}$$

At any time $t > 0$, the values of each node is in the form of $n\,a/2^b$ for some $a, b \in \mathbb{Z}^+ \cup \{0\}$. At consensus time t_c, we have $x_i(t_c) = 1$ so $n\,a/2^b = 1$ for some $a, b \in \mathbb{Z}^+ \cup \{0\}$. This means

$$n\,a = 2^b$$

which implies that n is a power of 2, contradicting the primality of n. □

From the proof of Lemma 7 we can see that any n which is not a power of 2 will not work with single edge consensus.

IV. k-EDGE SCHEDULES

Using the same approach as Lemma 7, we can get a necessary condition for finite time consensus:

Theorem 8 (Necessity)**.** *Let $p > 2$ be the largest prime that divides n. If $p > k+1$, then discrete time consensus is not possible.*

Proof. Suppose that finite time consensus is possible. Consider initial node values:

$$x_i(0) = \begin{cases} n & \text{if } i = 0 \\ 0 & \text{otherwise} \end{cases}$$

At any time $t > 0$,

$$x_i(t) = n \, \frac{a}{2^{j_2}\, 3^{j_3} \cdots k^{j_k}\, (k+1)^{j_{k+1}}}$$

for some $a \in \mathbb{Z}^+$ and $j_2, \ldots, j_{k+1} \in \mathbb{Z}^+ \cup \{0\}$. At consensus time, $x_i(t_c) = 1$ so we have

$$n\, a = 2^{j_2}\, 3^{j_3} \cdots k^{j_k}\, (k+1)^{j_{k+1}}$$

since p divides n, this implies that one of $\{2, 3, \ldots, k, k+1\}$ divides p, contradicting the primality of p. □

For sufficiency, let p be the largest prime that divides n with $p \leq k+1$ and consider the following scheduling algorithm that achieves consensus in finite time:

Algorithm 2: CONSENSUS

```
   Input: $k, \{x_0, x_1, \ldots, x_{n-1}\}$
   Output: For all $i$, $x_i = n^{-1} \sum_{j=0}^{n-1} x_j$
1  if $k \geq \binom{n}{2}$ then
2      Form $n$-clique with $\binom{n}{2}$ edges
3  else if $n \neq$ prime then
4      $q \leftarrow$ smallest prime factor of $n$
5      Index the nodes by $x_{i,j}$ where $0 \leq i < q$ and
       $0 \leq j < n/q$
6      for $i \leftarrow 0$ to $q-1$ do
7          Consensus($k, \{x_{i,0}, x_{i,1}, \ldots, x_{i,n/q-1}\}$)
8      end
9      for $j \leftarrow 0$ to $(n/q)-1$ do
10         Consensus($k, \{x_{0,j}, x_{1,j}, \ldots, x_{q-1,j}\}$)
11     end
12 else if $(k \geq n-1)$ and $(n = prime)$ then
13     Consensus($k, \{x_1, x_2, \ldots, x_{n-1}\}$)
14     Place $n-1$ edges between $x_0$ and $x_1$
15     Consensus($k, \{x_1, x_2, \ldots, x_{n-1}\}$)
16 else
17     Error: Finite time consensus not possible (Theorem
       8).
18 end
```

Correctness: The base case of $k \geq \binom{n}{2}$ (lines 1-3) is easy: we have enough edges to form a n-clique thus guaranteeing consensus in one step. Lines 3-11 represents a divide-and-conquer strategy: breaking down the problem into q subproblems of size n/q and n/q subproblems of size q. Since n is not prime, the division is always possible. This leaves us with lines 12-16. After execution of Line 13, the node values are of the form:

$$x_i = \begin{cases} a & \text{if } i = 0 \\ b & \text{if } 1 \leq i < n \end{cases}$$

for some $a, b \in \mathbb{R}$. After Line 14, we have

$$x_i = \begin{cases} \frac{1}{n}\,a + \frac{n-1}{n}\,b & \text{if } i = 0 \\ \frac{n-1}{n}\,a + \frac{1}{n}\,b & \text{if } i = 1 \\ b & \text{if } 2 \leq i < n \end{cases}$$

Notice that

$$\begin{aligned} \frac{1}{n-1}\sum_{i=1}^{n-1} x_i &= \frac{1}{n-1}\left(\frac{n-1}{n}\,a + \frac{1}{n}\,b + (n-2)\,b\right) \\ &= \frac{1}{n}\,a + \frac{n-1}{n}\,b \end{aligned}$$

so the node values are correct after Line 15.

Runtime: First, assume that we know the prime factorization of n so that Line 4 of the algorithm executes in $O(1)$ time. The solution to the recurrence

$$A(k) = 2\,A(k-1) + 1$$

is $A(k) = O(2^k)$. The solution to the recurrence

$$\begin{aligned} B(n) &= q\,B(n/q) + (n/q)\,B(q) \\ &\leq (n/q)\,B(n/q) \end{aligned}$$

is $B(n) = O(n \log n)$. These recurrences can be solved using any standard techniques (*e.g.* Chapter 4 of [22]).

To analyze Algorithm 2, we let $T(n)$ denote its runtime on n nodes. We have the following recurrence

$$T(n) = \begin{cases} 2\,T(n-1) + 1 & \text{if } n \leq k+1 \\ q\,T(n/q) + (n/q)\,T(q) & \text{otherwise} \end{cases}$$

Using solutions of $A(k)$ and $B(n)$. We have that

$$T(n) = O(2^k\, n \log n)$$

Even if we did not have the prime factorization of n, we can execute Line 4 in time $O(k)$. Because of Theorem 8, we only need to check n against all primes that are $\leq k+1$. This introduces a factor of k slow down and brings the run-time down to $O(k\, 2^k\, n \log n)$. To summarize everything:

Theorem 9. *Let $p > 2$ be the largest prime that divides n, then $t_c(\mathcal{S}_k) < \infty$ if and only if $p \leq k+1$. When it is finite*

$$t_c(\mathcal{S}_k) = O(n \log n)$$

Here, we treat k as a constant.

The exponential dependency on k is a result of Lines 12-15. Algorithm 2 proves the existence of finite time consensus schedules. We have not attempted to optimize it. Depending on the values of n and k, we can often remove the exponential dependence on k. More development on this subject can be found in the concluding discussions.

V. Discussion

Instead of asking for the minimum time required with a k-edge schedule, we can also ask about the edge requirements under a time constraint.

As an example, if we require that all nodes reach consensus in a single step, each node must have degree $n-1$. That is, a node must be able to see all other nodes. Since there are n nodes total, each with degree $n-1$, we require at least $n(n-1)/2 = \binom{n}{2}$ edges. This lowerbound is clearly feasible since a complete graph on n nodes achieves consensus in a single time step.

Lemma 10. *To reach consensus in a single step, we need* $\binom{n}{2}$ *edges.*

As stated previously, Algorithm 2 is not optimized. There are often scenarios where runtime can be greatly reduced. For example, in Section III, we showed matching upper and lower bounds for the special case of n being a power of 2. We can generalize this as follows:

Theorem 11. *If* $k = \binom{m}{2}$ *and* $n = m^q$ *for some* $q \in \mathbb{Z}^+$, *then*

$$t_c(\mathcal{S}_k) \le q\, m^{q-1} = \frac{n \log_m n}{m}$$

Assuming that the conditions in the Theorems 11 are true, the following scheduling algorithm achieves the upperbound:

```
Algorithm 3: CONSENSUS
   Input: $k, \{x_0, x_1, \ldots, x_{n-1}\}$
   Output: For all $i$, $x_i = n^{-1} \sum_{j=0}^{n-1} x_j$
1  if $k \ge \binom{n}{2}$ then
2      Form $n$-clique with $\binom{n}{2}$ edges
3  else
4      Index the nodes by $x_{i,j}$ where $0 \le i < n/m$ and $0 \le j < m$
5      for $i \leftarrow 0$ to $(n/m) - 1$ do
6          Consensus($k, \{x_{i,0}, x_{i,1}, \ldots, x_{i,m-1}\}$)
7      end
8      for $j \leftarrow 0$ to $m - 1$ do
9          Consensus($k, \{x_{0,j}, x_{1,j}, \ldots, x_{(n/m)-1,j}\}$)
10     end
11 end
```

Runtime: Since we have enough edges to form a m-clique, lines 5-7 takes n/m time steps. Let $T(n)$ denote the runtime of the algorithm on $n = m^q$ nodes, then we have the following recurrence

$$T(n) = \frac{n}{m} + m\, T\left(\frac{n}{m}\right)$$

which we can solve using any standard techniques (*e.g.* Chapter 4 of [22]) to show the correct runtime.

VI. Acknowledgment

The authors would like to acknowledge Xiaojie Gao, Richard Murray, Leonard Schulman, and Chris Umans for helpful discussions. We thank the anonymous reviewers whose comments improved the paper. We also thank one reviewer for pointing out an additional reference.

References

[1] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and Cooperation in Networked Multi-Agent Systems," *Proceedings of the IEEE, Special Issue on Networked Control Systems*, vol. 95, no. 1, pp. 215–233, 2007.

[2] S. Martínez, J. Cortés, and F. Bullo, "Motion coordination with distributed information," *IEEE Control Systems Magazine*, vol. 27, no. 4, pp. 75–88, 2007.

[3] R. Olfati-Saber and R. M. Murray, "Consensus Protocols for Networks of Dynamic Agents," in *American Control Conference*, 2003.

[4] J. A. Fax and R. M. Murray, "Information Flow and Cooperative Control of Vehicle Formations," *IEEE Trans. on Automatic Control*, vol. 49, pp. 1465–1476, September 2004.

[5] W. Xi, X. Tan, and J. S. Baras, "A Stochastic Algorithm for Self-Organization of Autonomous Swarms," in *IEEE Conference on Decision and Control*, 2005.

[6] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Distributed Sensor Fusion Using Dynamic Consensus," in *IFAC World Congress*, 2005.

[7] P. Ogren, E. Fiorelli, and N. E. Leonard, "Cooperative Control of Mobile Sensor Networks: Adaptive Gradient Climbing in a Distributed Environment," *IEEE Trans. on Automatic Control*, vol. 49, pp. 1292–1302, August 2004.

[8] T. Arai, E. Pagello, and L. E. Parker, "Guest Editorial Advances in Multirobot Systems," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 655 – 661, 2002.

[9] R. Olfati-Saber and R. M. Murray, "Consensus Problems in Networks of Agents with Switching Topology and Time-Delays," *IEEE Trans. on Automatic Control*, vol. 49, pp. 1520–2533, September 2004.

[10] W. Ren, R. W. Beard, and E. Atkins, "Information Consensus in Multivehicle Cooperative Control: Collective Group Behaviour Through Local Interaction," *IEEE Control Systems Magazine*, vol. 27, pp. 71–82, April 2007.

[11] L. Moreau, "Stability of continuous-time distributed consensus algorithms," in *IEEE Conference on Decision and Control*, pp. 3998 – 4003, 2004.

[12] R. Carli, F. Fagnani, A. Speranzon, and S. Zampieri, "Communication constraints in coordinated consensus problems," American Control Conference, June 2006.

[13] C. Godsil and G. Royle, *Algebraic Graph Theory*. Springer-Verlag, 2001.

[14] P. Yang, R. Freeman, and K. M. Lynch, "Optimal Information Propagation in Sensor Networks," in *International Conference on Robotics and Automation*, 2006.

[15] L. Xiao and S. Boyd, "Fast Linear Iterations for Distributed Averaging," *Systems and Control Letters*, vol. 53, pp. 65–78, September 2004.

[16] R. Olfati-Saber, "Ultrafast Consensus In Small-World Networks," in *American Control Conference*, 2005.

[17] J. Cortes, "Finite-time convergent gradient flows with applications to network consensus," *Automatica*, vol. 42, no. 11, pp. 1993 – 2000, 2006.

[18] L. Wang and F. Xiao, "Finite-time consensus problems for networks of dynamic agents," *http://arxiv.org/pdf/math/0701724*.

[19] S. Sundaram and C. N. Hadjicostis, "Finite-Time Distributed Consensus in Graphs with Time-Invariant Topologies," in *The 26th American Control Conference, New York, NY*, 2007.

[20] S. Sundaram and C. N. Hadjicostis, "Distributed consensus and linear functional calculation in networks: an observability perspective," in *The 6th International Conference on Information Processing in Sensor Networks*, 2007.

[21] T. M. Cover and J. A. Thomas, *Elements of information theory*. New York, NY, USA: Wiley-Interscience, 1991.

[22] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. The MIT Press, 2nd ed., 2001.