A One-Parameter Family of Distributed Consensus Algorithms with Boundary: From Shortest Paths to Mean Hitting Times

Alireza Tahbaz-Salehi and Ali Jadbabaie

Abstract— **We present a one-parameter family of consensus algorithms over a time-varying network of agents. The proposed family of algorithms contains the average and minimum consensus algorithms as two special cases. Furthermore, we investigate a closely related family of distributed algorithms which can be considered as a consensus scheme with fixed boundary conditions and constant inputs. The proposed algorithms recover both the Bellman-Ford iteration for finding shortest paths as well as the algorithm for calculating the mean hitting time of a random walk on a graph. Finally, we demonstrate the potential utility of these algorithms for routing in adhoc networks.**

I. Introduction

Recently, there has been a significant amount of attention on analysis and synthesis of scalable and provably correct algorithms for distributed motion coordination and control of multiagent systems. The main goal of such algorithms has been to achieve a certain global agreement in the multiagent system using purely local interactions. Simply put, a distributed consensus algorithm is an iterative (or dynamic in continuous time) scheme, according to which a set of agents exchange information with their nearest neighbors and perform a local computation to achieve asymptotic agreement among the agents' values. Due to absence of a centralized computational entity, possible lack of information on global topology and limited energy resources, one desirable property of such algorithms is robustness to occurrence of discontinuous changes in the topology of the multiagent network. These sudden and discontinuous changes in the network topology can occur naturally as a consequence of limited communication range between agents and motion.

While variants of such algorithms have an old history [1]–[3], they have recently resurfaced in the control literature in various contexts such as motion coordination and flocking of kinematic agents [4]–[6], computation of averages and least squares among sensors in a distributed fashion [7]–[9], and distributed locational and geometric optimization [10]. All of the above results, in one form or the other, represent a natural distributed implementation of an algorithm for solving a global optimization problem [3], [11]. In particular, many of such algorithms are distributed implementations of "discrete" analogues of certain PDEs such as Laplace's equation with no boundary [12] as well as with Dirichlet boundary conditions [13], [14].

This research is supported in parts by the following grants: ARO/MURI W911NF-05-1-0381, ONR/YIP N00014-04-1-0467 and NSF-ECS-0347285.

Alireza Tahbaz-Salehi and Ali Jadbabaie are with GRASP Laboratory and Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104-6228, USA. {atahbaz,jadbabai}@seas.upenn.edu

The goal of this paper is to develop a one-parameter family of general consensus algorithms which recovers both the average consensus [4], [7], [8] and the minimum consensus algorithms [9]. This one-parameter family of iterative algorithms utilizes a scaled version of the function that takes logarithm of a (weighted) average of exponentials of each node's value and its neighbors'. The function, known as the Log-Sum-Exp or LSE, appears in geometric programming problems and is often used as a smooth approximation of $\max$ and $\min$ functions [15]. However, when appropriately scaled by a parameter, the LSE function can also approximate the weighted average. As a result, one can use it in an iterative fashion with nearest neighbor computations in order to achieve consensus over the network. Another main contribution of this paper is to use the LSE function to recover two well-known algorithms as special cases, when a constant input and boundary conditions are added to the setup. We show that the LSE consensus algorithm recovers the Bellman-Ford iteration for computing shortest path lengths on one hand and the algorithm for computing the mean hitting time of a random walk on the other.

The paper is organized as follows: In section II we present the LSE consensus problem with minimum and average consensus as two special cases. Section III focuses on consensus problems with boundary conditions, and its connections to discrete Dirichlet problems. In section IV, we present a family of consensus algorithms with boundary conditions and inputs, as discrete analogues of a Poisson equation. Here we recover the mean hitting time algorithm as well as the Bellman-Ford scheme. We present a potential utility of these algorithms for routing in mobile adhoc networks in section IV-A. Finally, our conclusions are presented in section V.

II. Distributed Consensus Algorithms

Consider a group of n agents, labeled 1 through n, in the 2-dimensional space, each of which with an internal state that can be updated over time. We denote the internal state of agent i at time $t > 0$ by $y_i(t)$, and by $y_i(0)$ we mean its initial state value. Also assume that each agent has no information of the global topology of the network and is only capable of communicating with agents within a fixed distance (e.g. due to communication constraints such as power). We assume that this distance is the same for all nodes and equals to r. Therefore, agent j is capable of receiving agent i's state information at a given time if and only if the distance between the two is less than or equal to r at that time.

We further assume that the topology of the network changes at discrete time steps $t \in \{0, 1, 2, \cdots\}$ according to a piece-wise constant switching signal. At each time step t, we model the network topology by an undirected graph $\mathbb{G}(t) = (\mathcal{V}, \mathcal{E}(t))$. The vertex set of this graph, $\mathcal{V}$, is the set of all agents, and is fixed over time. The pair (i, j) is in $\mathcal{E}(t)$ if and only if the distance between nodes i and j is less than or equal to r at time t. In literature, such a graph is known as a *geometric graph* [16]. We use the notion $\mathcal{N}_i(t)$ to denote the set of neighbors of vertex i at time t.

In the rest of this section, we present a number of iterative update algorithms which eventually lead to a state value consensus over the network.

A. Average Consensus

First we explain a distributed iterative update scheme which results in the convergence of all the states to the average of the initial state values. Our scheme, in which the averaging matrix is chosen to be symmetric, is a special case of the results in [4] and [6] which only guarantee an asymptotic agreement on a *weighted* average (and not necessarily the average).

Our iterative update scheme is mainly based on the setup used in [8]. At each time step $t \geq 0$, the state of a node is updated as a weighted sum of the states of the neighboring nodes and its own,

$$y_i(t+1) = W_{ii}(t)y_i(t) + \sum_{j \in \mathcal{N}_i(t)} W_{ij}(t)y_j(t), \qquad (1)$$

where the values $W_{ij}(t)$ can be considered as positive weights at time t, satisfying $W_{ii}(t) = 1 - \sum_{j \in \mathcal{N}_i(t)} W_{ij}(t)$. By defining $W_{ij}(t) = 0$ for $j \notin \mathcal{N}_i(t) \cup \{i\}$, one can rewrite (1) in vector form as,

$$y(t+1) = W(t)y(t), \qquad (2)$$

where $y(t) = \left[y_1(t) \cdots y_n(t)\right]^T$ and matrix $W(t)$ has $W_{ij}(t)$ as its (i, j) entry. Now the question is whether (2) can result in an agreement in final value among all nodes, equal to the average of the initial state values. In other words, we are looking for sufficient conditions for

$$\lim_{t \to \infty} y(t) = \left(\frac{1}{n}\mathbf{1}^T y(0)\right)\mathbf{1} \qquad (3)$$

to hold. Here, $\mathbf{1}$ represents the vector with all entries equal to one. In order to guarantee convergence to the average, one can choose the weights to be symmetric, so that the averages are preserved over time. In [8], the authors suggest two weight matrices for this purpose. One is simply $W(t) = \frac{1}{n}(nI - D(t) + A(t))$, known as the maximum-degree weight matrix, in which $D(t) = \text{diag}(d_1(t), \cdots, d_n(t))$ and $A(t)$ are the valence and adjacency matrices of $\mathbb{G}(t)$, respectively. The other, Metropolis-Hastings, is defined as:

$$W_{ij}(t) = \begin{cases} \frac{1}{1+\max\{d_i(t), d_j(t)\}} & \text{if } (i,j) \in \mathcal{E}(t) \\ 1 - \sum\limits_{k \in \mathcal{N}_i} W_{ik}(t) & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

Before we state their results, we need to define the notion of *joint connectivity* of a collection of graphs. We say a finite collection of graphs $\{\mathbb{G}_1, \ldots, \mathbb{G}_m\}$ with common vertex set $\mathcal{V}$ is jointly connected if the union graph defined as $\mathbb{G} = (\mathcal{V}, \mathcal{E}_1 \cup \cdots \cup \mathcal{E}_m)$ is connected [4]. Also note that since there are only finitely many possible (geometric) graphs on n nodes, there exists at least one sequence of graphs that occurs infinitely often as time passes by. We have the following theorem.

Theorem 1: Suppose a jointly connected collection of graphs occurs infinitely often and the distributed averaging scheme uses either the maximum-degree or Metropolis-Hastings weights. Then (2) converges, and $\lim_{t \to \infty} y(t) = \frac{1}{n}\mathbf{1}\mathbf{1}^T y(0)$.

Proof: See [4], [6], [8] for details. ∎

This theorem shows that under the weak condition of joint connectivity, by using the given weight matrices for update scheme of (2), the states of all nodes converge to the average of the initial values. Note that this is not true for a general weight matrix. In fact, as shown in [4]–[6], although the network reaches a consensus using (2) with non-symmetric matrices, this common value may no longer be the average.

One can interpret the limit of the averaging iteration as the solution to the discrete counterpart of Laplace's equation $\nabla^2 y = 0$ in a continuous space. A solution to Laplace's equation is called a *harmonic function*. When there are no boundary conditions, any constant function satisfies Laplace's equation [14]. As a result, averaging based consensus algorithms are merely solving Laplace's equation over a discrete mesh.

B. Minimum Consensus

Now we explain another iterative process known as the minimum consensus. In this case the final state value of all nodes would be equal to the smallest initial value among them. Again consider the time-varying multiagent network with initial state value $y(0)$. At each following time step, each node updates its state to the minimum state value of the neighboring nodes and its own,

$$y_i(t+1) = \min\{y_i(t), \min_{j \in \mathcal{N}_i(t)} y_j(t)\}. \qquad (4)$$

For this update, we have the following:

Theorem 2: If a jointly connected collection of the graphs occurs infinitely often, then (4) terminates in finite number of steps and $\lim_{t \to \infty} y(t) = \min_{1 \leq i \leq n} y_i(0)\mathbf{1}$, leading to consensus.

Proof: For each node i, the sequence $\{y_i(t)\}_{t=0}^{\infty}$ is a non-increasing sequence and can only take values in a finite set. Therefore, the sequence converges and is constant for $t > T$, for some big enough T. Moreover, since there is a path from the minimum value node to any node i over time infinitely often, the limit of the sequence is the same for all nodes and equals to the minimum initial value. ∎

Note that the inclusion of the state value of node i in its own update formula is crucial to the convergence. Otherwise, the state values may oscillate and do not converge at all.

As another remark, notice that the theorem works under more relaxed conditions as well. Here, we used the strong notion of infinite often joint connectivity just in order to keep it consistent with other cases of consensus algorithms discussed. In the end, we point out that we will have similar results if we substitute $\min$ with $\max$ in (4), obtaining maximum consensus.

C. LSE Consensus

So far, we have presented two iterative processes which lead to a final value agreement over the network. The question is whether there exists a general class of algorithms, which contains average and minimum consensus algorithms as special cases. The following lemma suggests the use of Log-Sum-Exp functions for this purpose.

Lemma 1: The function $f_\gamma : \mathbb{R}^n \to \mathbb{R}$, defined as

$$f_\gamma(x) = \gamma \log \left(\frac{1}{n} \sum_{i=1}^{n} e^{x_i/\gamma} \right),$$

with parameter $\gamma \neq 0$, is (a) increasing in x with respect to the nonnegative orthant, (b) decreasing in γ over its continuity intervals, and (c) satisfies

$$\lim_{\gamma \to 0^-} f_\gamma(x) = \min\{x_1, \cdots, x_n\}, \quad (5)$$

$$\lim_{\gamma \to \infty} f_\gamma(x) = \frac{1}{n}(x_1 + \cdots + x_n). \quad (6)$$

Proof: (a) Suppose γ is positive. Also assume that $x \leq x'$ with respect to the nonnegative orthant, i.e. $x_i \leq x'_i$ for $1 \leq i \leq n$. Therefore, $\sum_{i=1}^n e^{x_i/\gamma} \leq \sum_{i=1}^n e^{x'_i/\gamma}$ holds. Since $\log$ is a monotone function, we get the desired result. The case of $\gamma < 0$ is similar. For (b), consider the function $g(\gamma) = \exp(f_\gamma(x))$ for a fixed x. The derivative of $g(\gamma)$ is

$$\frac{dg(\gamma)}{d\gamma} = \frac{g(\gamma)n}{\sum_i z_i} \left[\left(\frac{\sum_i z_i}{n} \right) \log \frac{\sum_i z_i}{n} - \frac{1}{n} \sum_i z_i \log z_i \right]$$

in which $z_i = e^{x_i/\gamma}$. Since $h(z) = z \log(z)$ is a convex function, the term inside the braces is negative. Therefore, $g(\gamma)$ and hence $f_\gamma(x)$ are decreasing functions of γ over continuity intervals. For the first limit in (c), if the smallest entry of x is $x_{\min}$, by factoring it out one can get,

$$f_\gamma(x) = x_{\min} + \gamma \log \left(\frac{1}{n} \sum_{i=1}^{n} e^{(x - x_{\min})/\gamma} \right).$$

Now, as $\gamma \to 0^-$, the term inside the $\log$ becomes $\frac{1}{n}$, and therefore, the function converges to the minimum entry of vector x. By changing 0^- with 0^+, the limit recovers the largest entry of x. To get (6), one can write the Taylor's expansion of the function and take its limit as $\gamma \to \infty$

$$\lim_{\gamma \to \infty} f_\gamma(x) = \lim_{\gamma \to \infty} \gamma \left(\frac{1}{\gamma} E(x) + \frac{1}{2\gamma^2} \mathit{Var}(x) \right) = E(x),$$

where $E(x) = \frac{1}{n} \sum_{i=1}^n x_i$ and $\mathit{Var}(x) = \frac{1}{n} \sum_{i=1}^n x_i^2 - E^2(x)$ are the sample mean and variance of the elements of x. ■

As the lemma shows, averaging and minimum operators are limit behaviors of LSE functions. This motivates us to define an iterative process based on these functions and obtain a one-parameter family of update algorithms. For this family, we set $y(0)$ as the initial state value and for $t \geq 0$,

$$y_i(t+1) = \gamma \log \left(\sum_{j=1}^{n} W_{ij}(t) e^{\frac{y_j(t)}{\gamma}} \right), \quad (7)$$

where the weights W_{ij} are defined similar to the averaging case earlier. By applying the variable change $z_i(t) = e^{y_i(t)/\gamma}$ and writing (7) in vector form, we get $z(t+1) = W(t)z(t)$, which is exactly as the averaging iterative process of (2). Therefore, by picking the weight matrix W to be either the maximum-wight matrix or Metropolis-Hastings, the convergence is guaranteed, and the limit is the average of the entries of vector $z(0)$, i.e. $\lim_{t \to \infty} z(t) = \frac{1}{n} \mathbf{1}\mathbf{1}^T z(0)$. Therefore,

$$\lim_{t \to \infty} y(t) = \gamma \log \left(\frac{1}{n} \sum_{j=1}^{n} e^{y_j(0)/\gamma} \right) \mathbf{1}.$$

This shows that all nodes reach an eventual agreement equal to the LSE of the initial values. Although the convergence is independent of parameter γ, the common limit of the state values depends on it. As we expected, this common limit becomes the average of the initial conditions as in (3), when $\gamma \to \pm\infty$. On the other hand, as $\gamma \to 0^-$, this common limit equals to the limit of the minimum consensus algorithm (4).

III. Consensus Algorithms with Boundary Conditions

In this section, we investigate the effect of adding boundary conditions to the iterative processes described earlier. By having boundary conditions, we mean the case in which one or more nodes in the network do not update their states, although their state values are used in the update process of their neighbors. One can interpret the boundary nodes of a network in different ways. For instance, the authors of [4], in which the state of each node represents the heading of an autonomous constant-speed agent, consider the boundary node as a group leader which keeps its heading constant.

Here, we mainly concentrate on the case of one boundary node and provide an example of the case with two such nodes. As described in the preceding section, average and minimum consensus algorithms are limit behaviors of the family of LSE iterative processes. Therefore, we first investigate the effect of boundary conditions on LSE family, and then on average and minimum updates as its limits.

Consider the time-varying network of n nodes, which at each time step can be represented by its corresponding time-dependent geometric graph, $\mathbb{G}(t) = (\mathcal{V}, \mathcal{E}(t))$. State of node i is initialized at some value $y_i(0)$ for $1 \leq i \leq n$. Another node, labeled $n+1$, with fixed state value y_c is added to the network. The addition of this boundary node (or leader as in [4]) is equivalent to adding a vertex to $\mathbb{G}(t)$ for all times. One can represent the new network topology with a modified geometric graph, $\tilde{\mathbb{G}} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}}(t))$. Here, $\tilde{\mathcal{V}} = \mathcal{V} \cup \{n+1\}$ is the vertex set and $\tilde{\mathcal{E}}(t) \supset \mathcal{E}(t)$ contains the adjacency information of the boundary node as well as of the others.

The LSE iterative process with parameter $\gamma \neq 0$ and one boundary node would be of the following form,

$$y_i(t+1) = \begin{cases} \gamma \log \left(\sum_{j=1}^{n+1} W_{ij}(t) e^{\frac{y_j(t)}{\gamma}} \right) & \text{if } i \neq n+1, \\ \\ y_i(t) & \text{if } i = n+1. \end{cases} \tag{8}$$

Here, unlike before $W(t) \in \mathbb{R}^{n \times (n+1)}$ is a matrix with unit row sums and satisfies $W_{ij}(t) \geq 0$ for $j \in \widetilde{\mathcal{N}}_i(t) \cup \{i\}$, and $W_{ij}(t) = 0$, otherwise. Clearly, the new value of node i only depends on the values of its neighbors. Therefore, (8) is a distributed iterative process for which the following theorem holds:

Theorem 3: Suppose there exists an infinite sequence of bounded, non-empty time intervals $[\tau_i, \tau_{i+1})$ with $\tau_0 = 0$, such that the collection $\{\mathbb{G}(\tau_i), \cdots, \mathbb{G}(\tau_{i+1} - 1)\}$ is jointly connected. Then (8) converges and $\lim_{t \to \infty} y(t) = y_c \mathbf{1}$.

The theorem says that the state values of all nodes in the network eventually converge to the value of the leader (boundary node), provided there is *enough* communication among them.

Proof: Apply the variable change $z_i(t) = e^{y_i(t)/\gamma}$. It would be sufficient to show $\lim_{t \to \infty} z(t) = z_{n+1}(0)\mathbf{1}$. Writing the update scheme in vector form leads to

$$z(t+1) = \widetilde{F}(t) z(t) = \begin{bmatrix} F(t) & b(t) \\ \mathbf{0} & 1 \end{bmatrix} z(t),$$

in which $F(t)$ is the principal submatrix of $W(t)$ of size n and $b = (I - F)\mathbf{1}$ is its last column. Therefore,

$$z(t) = \prod_{j=0}^{t-1} \widetilde{F}(j) z(0) = \begin{bmatrix} \Phi_{0t} & \sum_{j=1}^{t} \Phi_{jt} b(j-1) \\ \mathbf{0} & 1 \end{bmatrix} z(0)$$

where $\Phi_{tt} = I$ and $\Phi_{jt} = F(t-1) \cdots F(j)$. As proved in [4], [5], as long as there exists a jointly connected collection of graphs occurring infinitely often, we have

$$\lim_{t \to \infty} \Phi_{0t} = 0. \tag{9}$$

Since the matrix multiplied by $z(0)$ has unit row sums, under (9), its last column should converge to vector of ones. In other words, $\lim_{t \to \infty} \sum_{j=1}^{t} \Phi_{jt} b(j-1) = \mathbf{1}$, leading to the desired result. ■

As described earlier, average and minimum consensus algorithms are limiting behaviors of the family of LSE algorithms. For the case of averaging algorithm, the final common value can be regarded as the solution of the discrete counterpart of Laplace's equation $\nabla^2 y = 0$ with $\partial y = y_c$ as the boundary condition. The problem of finding the harmonic function y satisfying Laplace's equation with boundary conditions is known as the *Dirichlet problem*. One method for solving this problem is the method of relaxations, which is exactly the averaging iterative method [14].

The case for the minimum consensus algorithm with the presence of boundary conditions is slightly different. Similar to the no-boundary case, the minimum consensus algorithm does not converge in general if a node's value is not included in its own update. So, the inclusion of the node value in its update iteration as in

$$y_i(t+1) = \begin{cases} \min\{y_i(t), \min_{j \in \tilde{\mathcal{N}}_i} y_j(t)\} & \text{if } i \neq n+1 \\ 0 & \text{if } i = n+1 \end{cases}$$

is crucial to the convergence. But even then, although the iteration converges, it does not necessarily end up in a global consensus, unless the leader has the minimum value among all nodes. In fact, the nodes can only achieve local value agreements. More precisely, two non-boundary nodes reach the same value if there exists a path between the two not passing through the leader. Otherwise, they may have different values. The reason is that the leader blocks the propagation of information from one part of the graph to another.

We now investigate an example of the averaging algorithm with two boundary nodes which can be used in obtaining the *resistance distance* between nodes of the graph, in a distributed fashion. The resistance distance between a pair of vertices i and j, denoted by R_{ij}, is the electrical resistance measured across nodes i and j, when the network represents an electrical circuit with each edge as a unit resistor. It is known that the resistance distance between a pair of vertices is proportional to the expected *commute time* of the *natural random* walk (the random walk which jumps to a neighboring node with equal probability) between them [17], [18]. The commute time of a random walk on the graph between a pair of vertices i and j, is the time it takes to return to vertex i for the first time after starting from i and passing through vertex j [18].

Consider the time-invariant network of n nodes with two boundary nodes labeled $n+1$ and $n+2$ added to it. (This can be regarded as a special case of a more general framework in [13]). We denote the original n-vertex graph of the network by $\mathbb{G}$, and by $\tilde{\mathbb{G}}$ we are referring to the graph obtained by adding the two boundary nodes. The adjacency matrix of the latter can be written in terms of the adjacency matrix of the former,

$$\tilde{A} = \left[\begin{array}{c|cc} A & b_1 & b_2 \\ \hline b_1^T & c_{11} & c_{12} \\ b_2^T & c_{21} & c_{22} \end{array} \right]$$

in which $b_1, b_2 \in \mathbb{R}^{n \times 2}$ and $C \in \mathbb{R}^{2 \times 2}$. We set the values of nodes $n+1$ and $n+2$ to one and zero, respectively. If we denote the adjacency and valence matrices of $\mathbb{G}$ by A and D respectively, the averaging iterative update with the presence of these two boundary nodes and with uniform weights can be written in vector form as

$$y(t+1) = (D+B)^{-1} \left(A y(t) + \begin{bmatrix} b_1 & b_2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right),$$

where, $B = \text{diag}(b_1 + b_2)$ and $y(t)$ is the value of the n non-boundary nodes of the graph. Hence,

$$\begin{aligned} y(t) &= \left((D+B)^{-1} A \right)^t y(0) \\ &+ \sum_{k=1}^{t-1} \left((D+B)^{-1} A \right)^k (D+B)^{-1} \begin{bmatrix} b_1 & b_2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \end{aligned}$$

As proved in [13], the term $(D+B)^{-1}A$ has a subunit spectral norm. As a result, the first term on the right-hand-side goes to zero, and the second term converges, leading to

$$y^* = \lim_{t\to\infty} y(t) = (D+B-A)^{-1}\begin{bmatrix} b_1 & b_2 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Therefore, the limit y^* satisfies

$$(D+B-A)y^* = \begin{bmatrix} b_1 & b_2 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix}. \tag{10}$$

A reader familiar with circuit theory, can easily recognize (10) as Kirchhoff's current law (KCL) in the circuit obtained by replacing each edge of $\tilde{\mathbb{G}}$ with a unit resistance, connecting node $n+1$ to a unit voltage source and grounding $n+2$. Here, y^* denotes the voltage of the other n nodes of the graph. Therefore, the averaging algorithm is actually solving for the voltages of the nodes in the circuit. By having y^*, one can easily compute the current of each edge, in particular, the total current injected to the circuit by the voltage source. Hence, in this case, the averaging update can provide us with a method to compute the resistance distance between nodes $n+1$ and $n+2$, in a completely distributed fashion. Aside from this, y^* contains even more information about the topology of the graph. In fact, y_i^* is the probability that the natural random walk on the graph started from node i hits node $n+1$ before node $n+2$ [14], [17].

IV. Log-Sum-Exp Algorithms with Boundary Conditions and Input

In the previous sections we discussed the family of iterative LSE processes with and without the presence of boundary conditions. We also investigated the limit behaviors of this family, namely the averaging and minimum consensus algorithms. We now turn to another variant of these algorithms by introducing a fixed input to the update equation. We show that for the case of fixed network topology, the family of LSE algorithms can provide us with shortest path and mean hitting time information.

Consider the n-node network with initial state value $y(0)$. Also consider that a boundary node, labeled $n+1$, is added to the network, such that the overall resulting geometric graph, $\tilde{\mathbb{G}}$, is connected. We fix the state value of the boundary node to 0, and setup the following iterative process,

$$y_i(t+1) = \begin{cases} 1+\gamma \log\left(\sum_{j=1}^{n+1} W_{ij}(t) e^{\frac{y_j(t)}{\gamma}}\right) & \text{if } i \neq n+1 \\ 0 & \text{if } i = n+1. \end{cases} \tag{11}$$

Here, $W \in \mathbb{R}^{n\times(n+1)}$ is a row stochastic matrix with the sparsity pattern of the first n rows of $\tilde{A}+I$. We further assume that $\gamma < 0$ to guarantee that the iterative process converges. But before addressing its convergence, we investigate the limit behavior of (11) when γ varies. By lemma 1, when $\gamma \to 0^-$, the update in (11) becomes

$$y_i(t+1) = \begin{cases} 1+\min_{j\in\tilde{\mathcal{N}}_i} y_j(t) & \text{if } i \neq n+1 \\ 0 & \text{if } i = n+1. \end{cases} \tag{12}$$

The reader may realize that (12) is the Bellman-Ford iterative algorithm, which converges to the length of the shortest path between any node i and the boundary node $n+1$ over the graph [11]. In the context of routing on networks, if one regards node $n+1$ as the destination node, $\lim_{t\to\infty} y_i(t)$ would be the cost of routing a packet, in the sense of hop count, from node i to the destination node over the shortest path possible.

Now, we take a look at the other end of the spectrum, $\gamma \to -\infty$. In this case, the limit behavior of (11) would be

$$y_i(t+1) = \begin{cases} 1+\sum_{j\in\tilde{\mathcal{N}}_i} W_{ij} y_j(t) & \text{if } i \neq n+1 \\ 0 & \text{if } i = n+1. \end{cases} \tag{13}$$

One can regard this update as an iterative way to solve the discrete counterpart of *Poisson's equation*, $\nabla^2 y = u$, in a continuous space with some boundary condition [14]. Rewriting (13) in vector form leads to

$$y(t+1) = Fy(t) + \mathbf{1} = F^t y(0) + \sum_{k=0}^{t-1} F^k \mathbf{1},$$

in which $y(t)$ consists only of the first n node values (the last one is constantly zero), and F is the principal $n \times n$ submatrix of W. Since F is a substochastic square matrix,

$$\lim_{t\to\infty} y(t) = \sum_{k=0}^{\infty} F^k \mathbf{1} = (I-F)^{-1}\mathbf{1},$$

and (13) converges. The limit is the expected hitting time of node $n+1$ (or expected absorption time) of a random walk defined on the graph with $P(S_{t+1}=j|S_t=i) = W_{ij}$ as the transition probability, where S_t denotes the state of the random walk at time t [19], [20]. For the special case of uniform weights over the graph, i.e. treating all neighboring nodes equally, this limit would be the mean hitting time of the natural random walk defined on the graph. Similar to the Bellman-Ford case, in the context of routing one can regard node $n+1$ as the destination node. If so, then $\lim_{t\to\infty} y_i(t)$ is the expected cost of routing from node i, if the packet is sent randomly to an adjacent node at each time step.

Now, that we have seen the limiting interpretations of the LSE family, we proceed to a general member of this family. We have the following theorem.

Theorem 4: For $\gamma < 0$, the distributed iterative update of (11) converges and the limit does not depend on the initial values. Moreover, the limit is element-wise decreasing in γ.

Proof: Applying the variable change $z_i(t) = e^{y_i(t)/\gamma}$ and rewriting (11) in vector form leads to

$$z(t) = \begin{bmatrix} (e^{\frac{1}{\gamma}}F)^t & e^{\frac{1}{\gamma}} \sum_{k=0}^{t-1} (e^{\frac{1}{\gamma}}F)^k b \\ \mathbf{0} & 1 \end{bmatrix} z(0),$$

in which F is the $n \times n$ principal submatrix of W and $b = (I-F)\mathbf{1}$ is its last column. Since γ is a negative number, we have $\|e^{1/\gamma}F\| < \|F\| \leq 1$, guaranteeing convergence. By rewriting the equation in terms of y, we obtain the limit

$$y_\gamma^* = \lim_{t\to\infty} y(t) = \mathbf{1} + \gamma \mathbf{log}\left((I-e^{1/\gamma}F)^{-1}(I-F)\mathbf{1}\right),$$

which is independent of the initial state values. The proof of the second part is by induction. Consider two different setups of (11) with parameters $\gamma' < \gamma < 0$. Since the limit does not depend on the initial values, without loss of generality, one can assume that both iterations start from the same initial state values, $y(0) = y'(0)$. Now, as induction hypothesis, assume that $y(t) \leq y'(t)$. Therefore,

$$y(t+1) = f_\gamma(y(t)) \leq f_\gamma(y'(t)) \leq f_{\gamma'}(y'(t)) = y'(t+1).$$

Note that here, $f_\gamma : \mathbb{R}^n \to \mathbb{R}^n$ is a misuse of notation, and is referring to the update in (11). This result shows that $y(t)$ is less than or equal to $y'(t)$ for all t, and hence their limits satisfy the same ordering as well. ■

A. Discussion

The above theorem says that (11) provides us with a one-parameter family of converging algorithms. Intuitively, one can assume that there exists a trade-off between exploration and exploitation as γ varies between its two extremes. At one extreme, we have a natural random walk which deals indistinctively with all neighboring nodes. In this case, the random walk jumps to a neighboring node with equal probability and $y^*_{-\infty}$ provides us with the expected number of steps it takes for the walk to hit node $n+1$. On the other hand, one can interpret the other extreme as a random walk which, with probability one, moves towards the destination node, over the shortest path possible. Again, y^*_0 is the (expected) number of steps to hit $n+1$. For any other $\gamma \in (-\infty, 0)$, we have a combination of the two walks, and y^*_γ converges to a number between the two limits, providing us with some notion of expected number of steps to get to $n+1$. As a result, the scalar parameter γ can be interpreted as a "temperature" of some sort, controlling the entropy of the distribution.

As Bellman-Ford and its variants are a natural way of finding shortest paths and are extensively used for routing in networks, a good question to ask is whether other converged values can be used for routing as well. There is some evidence that this is indeed the case; using a simple counter-example argument one can show that in the limit, each node has a neighbor with a smaller limit value, since the iteration converges to a potential [14]. This suggests that the elements of vector y^*_γ can be used for routing over the network. The routing strategy would be as follows: if at time t the packet is in node i, it is transmitted to node $j \in \mathcal{N}_i$ at time $t+1$ such that $y^*_\gamma(j) \leq y^*_\gamma(k)$ for all $k \in \mathcal{N}_i$, until it eventually lands in the destination node. Based on the trade-off between exploration and exploitation, although paths derived for none zero values of γ are not necessarily optimal, we suspect them to be more robust to changes in the topology of the network.

V. Conclusion

In this paper, we presented a one-parameter family of consensus algorithms which recovers the minimum consensus and average consensus as two special cases. The key idea was to use the Log-Sum-Exp (LSE) function as the iteration function for each node and its neighbors. It was shown that modifications of the algorithm, by imposing a boundary node and adding an input, will result in a one-parameter family of iterative schemes that recovers the well-known Bellman-Ford iteration and the mean hitting time iterations as two special cases. We plan to extend these results by exploring the connections between this family of algorithms and Markov decision processes with entropy constraints in the future.

VI. Acknowledgements

The authors would like to thank Professor Stephen Boyd for helpful discussions.

References

[1] M. H. DeGroot, "Reaching a consensus," *Journal of American Statistical Association*, vol. 69, no. 345, pp. 118–121, Mar. 1974.
[2] J. N. Tsitsiklis, "Problems in decentralized decision making and computation," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 1984.
[3] N. A. Lynch, *Distributed Algorithms*. San Mateo, CA: Morgan Kaufmann, 1996.
[4] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.
[5] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 169–182, 2005.
[6] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference*, Seville, Spain, Dec. 2005, pp. 2996–3000.
[7] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time delays," *IEEE Transactions on Automatix Control*, vol. 49, no. 9, pp. 1520–1533, 2004.
[8] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proceedings of the 4th International Conference on Information Processing in Sensor Networks*, Los Angeles, CA, Apr. 2005, pp. 63–70.
[9] D. Bauso, L. Giarré, and R. Pesenti, "Distributed consensus in networks of dynamic agents," in *Proceedings of IEEE Conference on Decision and Control*, Seville, Spain, Dec. 2005, pp. 7054–7059.
[10] J. Cortes, S. Martinez, and F. Bullo, "Analysis and design tools for distributed motion coordination," in *Proceedings of the American Control Conference*, Portland, OR, June 2005, pp. 1680–1685.
[11] D. Bertsekas and J. Tsitsiklis, *Parallel and distributed computation*. Boston, MA: Athena Scientific, 1988.
[12] G. Ferrari-Trecate, A. Buffa, and M. Gati, "Analysis of coordination in multi-agent systems through partial difference equations. part i: The laplacian control," in *IFAC World Congress*, Prague, Czesh Republic, July 2005, electronic Proceedings.
[13] A. Jadbabaie, "On geographic routing without location information," in *Proceedings of IEEE Conference on Decision and Control*, Bahamas, 2004.
[14] P. G. Doyle and J. L. Snell, *Random Walks and Electric Networks*. Mathematical Association of America, 1984.
[15] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
[16] S. Boyd, A. Ghosh, B. Prabhakar, and D.Shah, "Mixing times for random walks on geometric random graphs," in *Proceedings of SIAM ANALCO*, 2005.
[17] D. Aldous and J. Fill, *Reversible Markov Chains and Random Walks on Graphs*, 2003, book in preperation. [Online]. Available: http://stat-www.berkeley.edu/users/aldous/RWG/book
[18] A. Ghosh, S. Boyd, and A. Saberi, "Minimizing effective resistance of a graph," in *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems*, Kyoto, Japan, July 2006, pp. 1185–1196.
[19] A. Platis, N. Limnois, and M. L. Du, "Hitting time in a finite nonhomogeneous markov chain with applications," *Applied Stochastic Models and Data Analysis*, vol. 14, no. 3, pp. 241–253, 1998.
[20] R. Durrett, *Probability: Theory and Examples*, 3rd ed. Belmont, CA: Duxbury Press, 2005.