

Reaching a Consensus with Limited Information

Jingxuan Zhu

Yixuan Lin

Ji Liu

A. Stephen Morse

Abstract—In its simplest form the well known consensus problem for a networked family of autonomous agents is to devise a set of protocols or update rules, one for each agent, which can enable all of the agents to adjust or tune their “agreement variable” to the same value by utilizing real-time information obtained from their “neighbors” within the network. The aim of this paper is to study the problem of achieving a consensus in the face of limited information transfer between agents. By this it is meant that instead of each agent receiving an agreement variable or real-valued state vector from each of its neighbors, it receives a linear function of each state instead. The specific problem of interest is formulated and provably correct algorithms are developed for a number of special cases of the problem.

I. INTRODUCTION

In its simplest form the well known consensus problem [2] for a networked family of autonomous agents is to devise a set of protocols or update rules, one for each agent, which can enable all of the agents to adjust or tune their “agreement variable” to the same value by utilizing real-time information obtained from their “neighbors” within the network. The consensus problem is one of the most fundamental problems in the area of distributed computation and control. Consensus algorithms can be found as components of a large variety of more specialized algorithms in the area of distributed computation and control such as distributed algorithms for solving linear algebraic equations [3], distributed optimization problems [4], distributed estimation problems [5], and even some distributed control problems [6].

There are a great many variations of the consensus problem. For example, the agreement variables could be restricted to be real-valued vectors or alternatively integer-valued vectors [7]. The updating of agreement variables could be executed either synchronously or asynchronously [8]. The topology of the network could be fixed or changing with time [9]. There could be malicious agents attempting to prevent consensus [10]. There could be communication delays [11] or bit-rate constraints [12]. The target value of the agreement variables could be unconstrained or it could

be some specified function of the initial values of the agents’ agreement variables as for example in distributed averaging [13] or gossiping [14]. Some versions of the problem such as when agreement variables take values in a finite set, defy deterministic solutions [7] whereas other versions of the problem do not.

The aim of this paper is to study the problem of achieving a consensus in the face of limited information transfer between agents. The problem setup is as follows. We consider a group of $m > 1$ autonomous agents labeled 1 to m . Each agent i has a set of neighbors from whom agent i can receive information; the set of labels of agent i ’s neighbors (excluding itself), denoted by $\mathcal{N}_i \subset \mathbf{m} \triangleq \{1, 2, \dots, m\}$, is part of the problem formulation. The neighbor sets $\mathcal{N}_i$, $i \in \mathbf{m}$, determine an m -vertex directed graph $\mathbb{N}$ defined so that there is an arc (or a directed edge) from vertex j to vertex i just in case agent j is a neighbor of agent i . Each agent i has an agreement variable or state $x_i \in \mathbb{R}^n$ which it can adjust synchronously at times $t \in \{0, 1, 2, \dots\}$. At time t , agent i receives from each neighbor $j \in \mathcal{N}_i$ a signal $s_{ji}(t) = C_{ji}x_j(t)$ where C_{ji} is a fixed real-valued matrix. Associating each arc (j, i) in $\mathbb{N}$ with matrix C_{ji} leads to a matrix-valued weighted neighbor graph $\bar{\mathbb{N}}$. It is assumed that for each $j \in \mathcal{N}_i$, $i \in \mathbf{m}$, both agents i and j know C_{ji} . There are no priori constraints on C_{ji} . Some could, for example, be matrices with less rows than columns in which cases the information transferred by each such corresponding signal $s_{ji}(t) = C_{ji}x_j(t)$ would be insufficient to determine $x_j(t)$. In this sense the information agent i receives from neighbor j at time t is limited to only a “part of” $x_j(t)$. Given this setup, the consensus problem of interest is to devise update rules using the $s_{ji}(t)$, one for each agent, which if possible will cause all m agents’ states x_i , $i \in \mathbf{m}$, to converge to the same value in the limit as $t \rightarrow \infty$.

II. WELL-CONFIGURED SYSTEMS

Consider the multi-agent system just described. We say that the m agents are in *local agreement* with specific states x_i , $i \in \mathbf{m}$, if $C_{ji}x_i = C_{ji}x_j$ for all $i \in \mathbf{m}$ and $j \in \mathcal{N}_i$. We say that the m agents have reached a *consensus* with specific states x_i , $i \in \mathbf{m}$, if $x_i = x_j$ for all $i, j \in \mathbf{m}$. A weighted neighbor graph $\bar{\mathbb{N}}$ is called *well-configured* if local agreement implies consensus.

A well-configured weighted neighbor graph $\bar{\mathbb{N}}$ has the following equivalent mathematical description. For each vertex i in $\mathbb{N}$, let d_i denote the number of neighbors of agent i . Then $d = \sum_{i=1}^m d_i$ equals the total number of directed edges in $\mathcal{E}$. Let $k_{i1}, \dots, k_{id_i}$ be an arbitrary ordering of the labels in $\mathcal{N}_i$. Label all the d arcs from 1 to d according to the sequence

Proofs of all assertions in this paper are omitted due to space limitations, which can be found in [1] and will be included in a journal version.

The work of J. Zhu, Y. Lin, and J. Liu was supported in part by the National Science Foundation under Grant No. 2230101 and by Stony Brook University’s Office of the Vice President for Research through a Seed Grant. The work of A.S. Morse was supported in part by the NSF under Grant No. 1917879 and by the AFOSR under Grant No. FA9550-20-1-0037.

J. Zhu and Y. Lin are with the Department of Applied Mathematics and Statistics at Stony Brook University ({jingxuan.zhu,yixuan.lin.1}@stonybrook.edu). J. Liu is with the Department of Electrical and Computer Engineering at Stony Brook University (ji.liu@stonybrook.edu). A.S. Morse is with the Department of Electrical Engineering at Yale University (as.morse@yale.edu).

$k_{11}, \dots, k_{1d_1}, \dots, k_{m1}, \dots, k_{md_m}$. Define the corresponding incidence matrix J as an $m \times d$ matrix in which column k has exactly one 1 in row i and exactly one -1 in row j if the k th arc in $\mathbb{N}$ is (j, i) . For any finite set of matrices $\{M_1, M_2, \dots, M_k\}$, we use $\text{blockdiag}\{M_1, M_2, \dots, M_k\}$ to denote the block diagonal matrix whose i th diagonal block is M_i . Define

$$C = \text{blockdiag}\{C_{k_{11},1}, \dots, C_{k_{1d_1},1}, \dots, C_{k_{m1},m}, \dots, C_{k_{md_m},m}\}.$$

Let $\bar{J} = J \otimes I_n$ and $\bar{I} = \mathbf{1}_m \otimes I_n$, where $\otimes$ denotes the Kronecker product, I_n denotes the $n \times n$ identity matrix, and $\mathbf{1}_m$ denotes the m -dimensional column vector whose entries all equal 1. Then it is not hard to verify that a weighted neighbor graph $\bar{\mathbb{N}}$ is well-configured if and only if

$$\text{kernel } C\bar{J}' = \text{span } \bar{I}. \quad (1)$$

In the case when $\mathbb{N}$ is weakly connected, $\text{kernel } \bar{J}' = \text{span } \bar{I}$ [15, Theorem 8.3.1]; then (1) will be true if and only if

$$\text{span } \bar{J}' \cap \text{kernel } C = 0. \quad (2)$$

It is worth emphasizing that C and J are defined according to the same ordering of the arcs in $\bar{\mathbb{N}}$, and the necessary and sufficient condition (1) or (2) is independent of the ordering.

With the above in mind, the following two questions arise. First, what are the necessary and/or sufficient conditions on $\mathbb{N}$ for which there exist C_{ji} matrices so that $\bar{\mathbb{N}}$ is well-configured? Second, if $\bar{\mathbb{N}}$ is well-configured, how one can construct a recursive distributed algorithm for each agent which will drive the system from arbitrary start states to local agreement and thus to a consensus? These are precisely what we consider in this paper.

III. SYSTEM DESIGN

The goal of this section is to derive graph-theoretic conditions on which a multi-agent system can be well-configured.

As described, for any pair of neighboring agents, say agent i and its neighbor j , agent j only sends $C_{ji}x_j$ to agent i so that the transmitted vector size may be reduced and x_j may not be identified. Thus it is sometimes desirable that $\mathcal{K}_{ji} \neq 0$, where $\mathcal{K}_{ji}$ denotes the kernel of C_{ji} ; otherwise, x_j can be uniquely determined from $C_{ji}x_j$. Also, if $\mathcal{K}_{ji} \neq 0$, the size of $C_{ji}x_j$ will be no smaller than that of x_j .

A directed graph $\mathbb{G}$ is called rooted if it contains a directed spanning tree of $\mathbb{G}$, and called strongly connected if there is a directed path between each pair of distinct vertices. Every strongly connected graph is rooted, but not vice versa.

First, it is easy to see that if $\mathbb{N}$ is not rooted, a consensus cannot be guaranteed for arbitrary initial values. We next consider some examples of rooted graphs.

A. Rooted Graphs

If $\mathbb{N}$ is rooted, $\bar{\mathbb{N}}$ cannot be always well-configured with all $\mathcal{K}_{ji} \neq 0$, as shown in the following lemma for path graphs.

Lemma 1: If $\mathbb{N}$ is a directed path, then $\bar{\mathbb{N}}$ can be well-configured only if all $\mathcal{K}_{ji} = 0$.

There exists a rooted graph which can be well-configured with all $\mathcal{K}_{ij} \neq 0$; see Example 1 in [1].

It turns out that well-configuration characterization of rooted graphs is quite complicated. We thus leave it as a future direction and focus on strongly connected graphs in the next subsection.

B. Strongly Connected Graphs

Strong connectedness itself cannot guarantee well-configuration. To state our sufficient condition for well-configuration, we need the following concept from graph theory [16].

An *ear decomposition* of a directed graph without self-arcs¹ $\mathbb{G} = (\mathcal{V}, \mathcal{E})$ with at least two vertices is a sequence of subgraphs of $\mathbb{G}$, denoted $\{\mathbb{E}_0, \mathbb{E}_1, \dots, \mathbb{E}_p\}$, in which $\mathbb{E}_0$ is a directed cycle, and each $\mathbb{E}_i$, $i \in \mathbf{p}$, is a directed path or a directed cycle with the following properties:

- 1) $\{\mathbb{E}_0, \mathbb{E}_1, \dots, \mathbb{E}_p\}$ form an arc partition of $\mathbb{G}$, i.e., $\mathbb{E}_i$ and $\mathbb{E}_j$ are arc disjoint if $i \neq j$, and $\bigcup_{k=0}^p \mathbb{E}_k = \mathbb{G}$;
- 2) For each $i \in \mathbf{p}$, if $\mathbb{E}_i$ is a directed cycle, then it has precisely one vertex in common with $\bigcup_{k=0}^{i-1} \mathbb{E}_k$; if $\mathbb{E}_i$ is a directed path, then its two end-vertices are the only two vertices in common with $\bigcup_{k=0}^{i-1} \mathbb{E}_k$.

Each of $\mathbb{E}_0, \mathbb{E}_1, \dots, \mathbb{E}_p$ is called an *ear* of the decomposition. Not all directed graphs admit an ear decomposition. It has been proved that a directed graph has an ear decomposition if and only if it is strongly connected [16, Theorem 7.2.2]. It is also known that there exists a linear algorithm to find one ear decomposition of a strongly connected graph [16, Corollary 7.2.5]. A strongly connected graph may admit multiple ear decompositions, and apparently, the number of all possible different ear decompositions of a strongly connected graph is finite. It turns out that every ear decomposition of a strongly connected graph with m vertices and e arcs has $e - m + 1$ ears [16, Corollary 7.2.3]. To help understand the concept, an illustrative example is provided in Figure 1.

Two subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$ of $\mathbb{R}^n$ are independent if their intersection is the zero subspace, i.e., if $\mathcal{S}_1 \cap \mathcal{S}_2 = 0$. A finite family of subspaces $\{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_p\}$ is independent if

$$\mathcal{S}_i \cap \left(\sum_{j \neq i} \mathcal{S}_j \right) = 0, \quad i \in \mathbf{p}.$$

Theorem 1: Suppose that $\mathbb{N}$ is strongly connected and let D be an ear decomposition of $\mathbb{N}$. If for each ear $\mathbb{E} \in D$, $\{\mathcal{K}_{ji} : (j, i) \in \mathbb{E}\}$ is an independent family, then $\bar{\mathbb{N}}$ is well-configured.

To prove the theorem, we first study directed cycles and paths since they are basic components in ear decompositions.

To simplify notation, we label the vertices of an m -vertex directed cycle as $1 \rightarrow 2 \rightarrow \dots \rightarrow m \rightarrow 1$. Suppose that $C_1, C_2, \dots, C_m$ are given matrices, each with n columns. Suppose that for each $i \in \mathbf{m}$, agent i receives $C_i x_{i-1}$ from agent $i-1$, where it is understood that agent 0 and agent m

¹The definition can be extended to more general directed multigraphs with self-arcs [16].

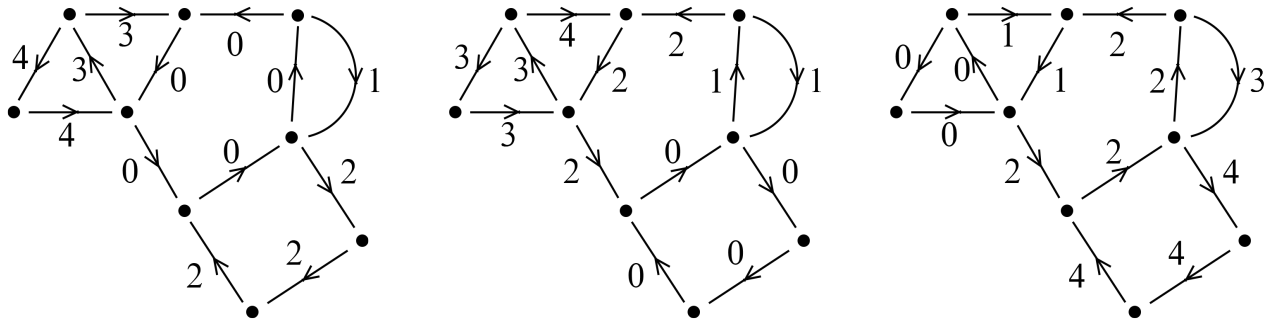


Fig. 1: An example of different ear decompositions of a strongly connected graph. The number associated to each arc represents the index of the ear which the arc belongs to in an ear decomposition.

are one and the same, and that $x_0 \triangleq x_m$. Thus for this $\bar{\mathbb{N}}$ to be well-configured means that the relations

$$C_i x_i = C_i x_{i-1}, \quad i \in \mathbf{m}, \quad (3)$$

must imply that $x_i = x_{i-1}$, $i \in \mathbf{m}$. Let $\mathcal{K}_i$ denote the kernel of C_i for all $i \in \mathbf{m}$.

Lemma 2: If $\mathbb{N}$ is an m -vertex directed cycle, then $\bar{\mathbb{N}}$ is well-configured by matrices C_i , $i \in \mathbf{m}$, if and only if $\{\mathcal{K}_1, \mathcal{K}_2, \dots, \mathcal{K}_m\}$ is an independent family.

It is easy to see that n is the maximum possible number of subspaces in an independent family of nonzero subspaces of $\mathbb{R}^n$. We thus have the following immediate consequence of Lemma 2.

Corollary 1: If $\mathbb{N}$ is an m -vertex directed cycle, then $\bar{\mathbb{N}}$ can be well-configured with all $\mathcal{K}_i \neq 0$, $i \in \mathbf{m}$, if and only if $m \leq n$.

More can be said.

Lemma 3: Let $\mathbb{N}$ be an m -vertex directed cycle with edge set $\mathcal{E}_{\mathbb{N}}$. Let $\mathcal{E}$ be a subset of $\mathcal{E}_{\mathbb{N}}$ defined as $\mathcal{E} = \{(i, j) \in \mathcal{E}_{\mathbb{N}} : x_i = x_j\}$. Then $\bar{\mathbb{N}}$ is well-configured by matrices C_i , $i \in \mathbf{m}$, if and only if $\{\mathcal{K}_i : i \in \mathbf{m}, (i-1, i) \notin \mathcal{E}\}$ is an independent family.

Lemma 3 immediately implies the following result.

Corollary 2: Let $\mathbb{N}$ be an m -vertex directed cycle with edge set $\mathcal{E}_{\mathbb{N}}$. Let $\mathcal{E}$ be a subset of $\mathcal{E}_{\mathbb{N}}$ defined as $\mathcal{E} = \{(i, j) \in \mathcal{E}_{\mathbb{N}} : x_i = x_j\}$. Then $\bar{\mathbb{N}}$ can be well-configured with all $\mathcal{K}_i \neq 0$ if and only if $m - |\mathcal{E}| \leq n$.

The above results can be directly applied to the following special case of path graphs.

To simplify notation, we label the vertices of an m -vertex directed path as $1 \rightarrow 2 \rightarrow \dots \rightarrow m$. Suppose that $C_2, \dots, C_m$ are given matrices, each with n columns. Suppose that for each $i \in \mathbf{m}$, agent i receives $C_i x_{i-1}$ from agent $i-1$. Thus for this $\bar{\mathbb{N}}$ to be well-configured means that the relations $C_i x_i = C_i x_{i-1}$, $i \in \{2, \dots, m\}$, must imply that $x_i = x_{i-1}$, $i \in \{2, \dots, m\}$. Adding the arc $(m, 1)$ to the above path and imposing $x_1 = x_m$ will lead to a special case satisfying the condition in Lemma 3 and Corollary 2, which immediately implies the following result.

Corollary 3: If $\mathbb{N}$ is an m -vertex directed path with $x_1 = x_m$, then $\bar{\mathbb{N}}$ is well-configured by matrices C_i , $i \in \{2, \dots, m\}$, if and only if $\{\mathcal{K}_2, \dots, \mathcal{K}_m\}$ is an independent family, and thus $\bar{\mathbb{N}}$ can be well-configured with all $\mathcal{K}_i \neq 0$, $i \in \{2, \dots, m\}$, if and only if $m-1 \leq n$.

Compared with Lemma 1, it is worth emphasizing that assuming $x_1 = x_m$ significantly changes the condition for well-configuration of path graphs.

The proof of Theorem 1 can be found in [1] and provides a constructive approach that systematically designs C_{ji} matrices for a strongly connected multi-agent system to be well-configured.

For each ear decomposition, say $\mathcal{D} = \{\mathbb{E}_0, \mathbb{E}_1, \dots, \mathbb{E}_p\}$, let $l(\mathbb{E}_i)$ denote the length of ear $\mathbb{E}_i$, i.e., the number of arcs in $\mathbb{E}_i$. Theorem 1 immediately implies the following sufficient conditions for well-configuration.

Corollary 4: Suppose that $\mathbb{N}$ is strongly connected and let $\mathcal{D}$ be an ear decomposition of $\mathbb{N}$. If $\max_{\mathbb{E} \in \mathcal{D}} l(\mathbb{E}) \leq n$, then $\bar{\mathbb{N}}$ can be well-configured with all $\mathcal{K}_{ji} \neq 0$, $i \in \mathbf{m}$, $j \in \mathcal{N}_i$.

More can be said. For a strongly connected graph $\mathbb{G}$, write $\mathcal{D}$ for the set of all possible ear decompositions of $\mathbb{G}$. Define

$$\chi(\mathbb{G}) = \min_{\mathcal{D} \in \mathcal{D}} \max_{\mathbb{E} \in \mathcal{D}} l(\mathbb{E}).$$

Since each ear decomposition begins with a directed cycle and the shortest possible length of a cycle is two, e.g., a pair of agents which are neighbors of each other, $\chi(\mathbb{G}) \geq 2$.

Corollary 5: If $\mathbb{N}$ is strongly connected and $\chi(\mathbb{N}) \leq n$, then $\bar{\mathbb{N}}$ can be well-configured with all $\mathcal{K}_{ij} \neq 0$, $i \in \mathbf{m}$, $j \in \mathcal{N}_i$.

Although Corollary 5 provides a weaker condition, to our knowledge, it is still an open problem to construct an efficient algorithm to find all ear decompositions of a strongly connected graph.

C. Symmetric Directed Graphs

A directed graph is called *symmetric* if whenever (i, j) is an arc in the graph, so is (j, i) . A symmetric directed graph is often called undirected in the literature, which simplifies each pair of directed edges, say (i, j) and (j, i) , to one undirected edge between vertices i and j . We stick to the

term “symmetric directed graphs” because of definition of the incidence matrix given in Section II. Consider a symmetric directed graph with m vertices and d directed edges. Then d must be an even number. Our definition of an incidence matrix is of size $m \times d$, while the standard definition of an incidence matrix of the corresponding undirected graph is of size $m \times (d/2)$. Thus using the term “undirected” may cause confusion. It is worth noting that rooted and strong connectedness boil down to the same connectivity for symmetric directed graphs.

For any symmetric directed graph $\mathbb{G}$, since each pair of arcs between any pair of neighboring agents in a symmetric directed graph is a cycle with length 2, all these cycles form an ear decomposition, which leads to $\chi(\mathbb{G}) = 2$. The following necessary and sufficient condition on well-configuration for symmetric directed graphs is easy to derive from Corollary 5.

Theorem 2: If $\mathbb{N}$ is a symmetric directed graph, then $\bar{\mathbb{N}}$ can be well-configured with all $\mathcal{K}_{ji} \neq 0$, $i \in \mathbf{m}$, $j \in \mathcal{N}_i$, if and only if $\mathbb{N}$ is strongly connected and $n \geq 2$.

As will be seen in the next section, there is a motivation, for the purpose of algorithm design, to figure out a condition under which a symmetric directed graph can be well-configured with the additional constraint that $C_{ij} = C_{ji}$ for all $i \in \mathbf{m}$ and $j \in \mathcal{N}_i$. To this end, we need the following modified concept of ear decompositions.

A *symmetric ear decomposition* of a symmetric directed graph without self-arcs $\mathbb{G} = (\mathcal{V}, \mathcal{E})$ with at least two vertices is a sequence of symmetric subgraphs of $\mathbb{G}$, denoted $\{\mathbb{E}_0, \mathbb{E}_1, \dots, \mathbb{E}_p\}$, in which $\mathbb{E}_0$ is a symmetric directed cycle, and each $\mathbb{E}_i$, $i \in \mathbf{p}$, is a symmetric directed path or a symmetric directed cycle with the following properties:

- 1) $\{\mathbb{E}_0, \mathbb{E}_1, \dots, \mathbb{E}_p\}$ form an arc partition of $\mathbb{G}$, i.e., $\mathbb{E}_i$ and $\mathbb{E}_j$ are arc disjoint if $i \neq j$, and $\bigcup_{k=0}^p \mathbb{E}_k = \mathbb{G}$;
- 2) For each $i \in \mathbf{p}$, if $\mathbb{E}_i$ is a symmetric directed cycle, then it has precisely one vertex in common with $\bigcup_{k=0}^{i-1} \mathbb{E}_k$; if $\mathbb{E}_i$ is a symmetric directed path, then its two end-vertices are the only two vertices in common with $\bigcup_{k=0}^{i-1} \mathbb{E}_k$.

Each of $\mathbb{E}_0, \mathbb{E}_1, \dots, \mathbb{E}_p$ is called a *symmetric ear* of the decomposition. Not all symmetric directed graphs admit a symmetric ear decomposition. A symmetric directed graph is called *k-connected* if, upon removal of any $k-1$ two-length cycles, the resulting graph is still strongly connected. It has been proved that a symmetric directed graph has a symmetric ear decomposition if and only if it is 2-connected [17].² A 2-connected symmetric directed graph may admit multiple symmetric ear decompositions, and apparently, the number of all possible different symmetric ear decompositions is finite. For each symmetric ear decomposition, say $\mathbf{D} = \{\mathbb{E}_0, \mathbb{E}_1, \dots, \mathbb{E}_p\}$, let $l(\mathbb{E}_i)$ denote the length of symmetric ear $\mathbb{E}_i$, i.e., the number of two-length cycles in $\mathbb{E}_i$. Using

²This is because a symmetric ear decomposition of a symmetric directed graph is essentially equivalent to an ear decomposition of an undirected graph, and a k -connected symmetric directed graph is essentially equivalent to a k -edge-connected undirected graph.

the same arguments as in the proof of Theorem 1, we have the following result.

Theorem 3: Suppose that $\mathbb{N}$ is 2-connected symmetric directed graph and let $\mathbf{D}$ be a symmetric ear decomposition of $\mathbb{N}$. If for each symmetric ear $\mathbb{E} \in \mathbf{D}$, $\{\mathcal{K}_{ij}f \vee \mathcal{K}_{ji} : (i, j) \in \mathbb{E}\}$ ³ is an independent family, then $\bar{\mathbb{N}}$ is well-configured by matrices $C_{ij} = C_{ji}$, $i \in \mathbf{m}$, $j \in \mathcal{N}_i$. If, in addition, $\max_{\mathbb{E} \in \mathbf{D}} l(\mathbb{E}) \leq n$, then $\bar{\mathbb{N}}$ can be well-configured with all $\mathcal{K}_{ij} = \mathcal{K}_{ji} \neq 0$, $i \in \mathbf{m}$, $j \in \mathcal{N}_i$.

In the sequel, we will propose and analyze a few distributed algorithms for well-configured systems under different scenarios.

IV. ALGORITHMS FOR SYMMETRIC DIRECTED GRAPHS

In this section, we assume that the neighbor graph is symmetric and $C_{ij} = C_{ji}$ whenever agents i and j are a pair of neighbors. We begin with the simplest case in which the neighbor graph is fixed.

A. Fixed Symmetric Directed Graphs

Consider any strongly connected, symmetric directed graph $\mathbb{N}$ with m agents. Our first algorithm appeals to the idea of gradient descent in convex optimization, which is for each agent i ,

$$x_i(t+1) = x_i(t) - \alpha(t) \sum_{j \in \mathcal{N}_i} \left[(C'_{ij}C_{ij} + C'_{ji}C_{ji}) \times (x_i(t) - x_j(t)) \right], \quad (4)$$

where $\alpha(t)$ is a positive time-varying stepsize satisfying $\sum_t \alpha(t) = \infty$ and $\sum_t \alpha^2(t) < \infty$.

Theorem 4: If $\mathbb{N}$ is a strongly connected symmetric directed graph and $\bar{\mathbb{N}}$ is well-configured, then algorithm (4) will lead all the agents to reach a consensus.

The algorithm (4) involves a term $(C'_{ij}C_{ij} + C'_{ji}C_{ji})(x_i(t) - x_j(t))$ in each agent i 's update, where j is any neighbor of agent i . In the case when $C_{ij} \neq C_{ji}$, it will require that each agent i receives two signals, $C_{ji}x_j(t)$ and $C_{ij}x_j(t)$, from each of its neighbors at each time step. Although allowing $C_{ij} \neq C_{ji}$ in a symmetric directed graph makes well-configuration easier in light of Theorem 2, transmitting two signals could be an issue in communication. In the case when $C_{ij} = C_{ji}$ so that only one signal is transferred, the underlying symmetric directed graph will need to be 2-connected to guarantee well-configuration. These facts are true for all the remaining algorithms in this section.

The above algorithm requires all m agents share the same sequence of diminishing stepsizes. Our second algorithm gets around this limitation and is thus fully distributed, which is described as follows.

Since well-configuration only depends on $\mathcal{K}_{ij}$, the kernel of C_{ij} , $i \in \mathbf{m}$, $j \in \mathcal{N}_i$, without loss of generality, we assume each C_{ij} has full row rank and its rows are

³We use $\{a \vee b\}$ to denote that either a or b is an element in the set.

orthonormal, which implies that $C_{ij}C'_{ij} = I$ and $P_{ij} \triangleq C'_{ij}(C_{ij}C'_{ij})^{-1}C_{ij} = C'_{ij}C_{ij}$ is an orthogonal projection matrix. For each agent $i \in \mathbf{m}$,

$$x_i(t+1) = x_i(t) - \frac{1}{2(d_i+1)} \sum_{j \in \mathcal{N}_i} \left[(C'_{ij}C_{ij} + C'_{ji}C_{ji}) \times (x_i(t) - x_j(t)) \right]. \quad (5)$$

Theorem 5: If $\mathbb{N}$ is symmetric, strongly connected and $\bar{\mathbb{N}}$ is well-configured, then algorithm (5) will lead all the agents to reach a consensus exponentially fast.

To prove the theorem, we need the following lemmas.

Lemma 4: If $\bar{\mathbb{N}}$ is well-configured, then $\bar{J}C'C\bar{J}'$ is positive semidefinite with exactly m eigenvalues at zero.

Lemma 5: Let $\bar{W} = W \otimes I$, where W is a positive diagonal matrix. If $\bar{\mathbb{N}}$ is well-configured, then $\bar{W}\bar{J}C'C\bar{J}'$ has exactly n eigenvalues at zero, and all the remaining eigenvalues are positive.

B. Time-varying Symmetric Directed Graphs

In this subsection, we consider the following scenario of time-varying symmetric directed graphs. Let an m -vertex symmetric directed graph $\mathbb{N}$ represent all allowable communication among the m agents. In other words, agents i and j are allowed to communicate with each other if and only if (i, j) is an arc in $\mathbb{N}$. For each time t , we use a time-dependent m -vertex symmetric directed graph $\mathbb{N}(t)$ to describe the neighbor relations among the m agents at time t . That is, if agents i and j communicate at time t , then (i, j) is an arc in $\mathbb{N}(t)$. It is easy to see that $\mathbb{N}(t)$ is a spanning subgraph of $\mathbb{N}$, and all such possible spanning subgraphs is a finite set. We assume that $\bar{\mathbb{N}}$ is well-configured, i.e., each arc (i, j) in $\mathbb{N}$ is associated with a matrix C_{ij} such that kernel $C\bar{J}' = \text{span } \bar{I}$, with J being the incidence matrix of $\mathbb{N}$.

For any time-varying symmetric directed graph sequence just described, we propose the following algorithm using the Metropolis weights:

$$x_i(t+1) = x_i(t) - \frac{1}{2} \sum_{j \in \mathcal{N}_i(t)} \left[w_{ij}(t)(C'_{ij}C_{ij} + C'_{ji}C_{ji}) \times (x_i(t) - x_j(t)) \right], \quad (6)$$

where $\mathcal{N}_i(t)$ is the neighbor set of agent i at time t and $w_{ij}(t)$ are the Metropolis weights corresponding to $\mathbb{N}(t)$, which are proposed in [18] for solving the distributed averaging problem over symmetric directed graphs and defined as

$$w_{ij}(t) = \frac{1}{1 + \max\{d_i(t), d_j(t)\}}, \quad j \in \mathcal{N}_i(t),$$

where $d_i(t) = |\mathcal{N}_i(t)|$ denotes the number of neighbors of agent i at time t .

Theorem 6: Suppose that $\bar{\mathbb{N}}$ is well-configured. If $\mathbb{N}$ is symmetric, strongly connected and each edge of $\mathbb{N}$ appears infinitely often in the infinite sequence of neighbor graphs

$\mathbb{N}(1), \mathbb{N}(2), \mathbb{N}(3), \dots$, then algorithm (6) will guarantee all m agents to reach a consensus.

To prove the theorem, we first combine the m update equations in (6) into one state form. To this end, we tailor the definition of an incidence matrix for spanning subgraphs as follows. Consider a directed graph $\mathbb{G}$ with m vertices and d directed edges. Let $\mathcal{E}$ denote the arc set of $\mathbb{G}$ and J denote the $m \times d$ incidence matrix of $\mathbb{G}$ according to some ordering of the arcs in $\mathcal{E}$. Let $\mathbb{H}$ be a spanning subgraph of $\mathbb{G}$. We define the *spanning incidence matrix* of $\mathbb{H}$ as an $m \times d$ matrix in which column k has exactly one 1 in row i and exactly one -1 is row j if the k th arc in $\mathbb{G}$ is (j, i) and (j, i) is also an arc in $\mathbb{H}$. It is clear that the spanning incidence matrix of any spanning subgraph of $\mathbb{G}$ has the same size as the incidence matrix of $\mathbb{G}$. If the k th arc in $\mathbb{G}$ is not in a spanning subgraph, then the k th column of the incidence matrix of $\mathbb{G}$ is replaced by a zero vector in the spanning incidence matrix.

We also need the following definition. Consider a symmetric directed graph $\mathbb{G}$ with d arcs. Let $\bar{\mathbb{G}}$ be a spanning subgraph of $\mathbb{G}$ which is also symmetric. Since $\bar{\mathbb{G}}$ is symmetric, its Metropolis weights $\bar{w}_{ij}$ are well-defined; specifically, $\bar{w}_{ij} = 1/(1 + \max\{\bar{d}_i, \bar{d}_j\})$, where $\bar{d}_k$ denotes the number of neighbors of vertex k in $\bar{\mathbb{G}}$. Given an ordering of all the arcs in $\mathbb{G}$, the *spanning weight matrix* of $\bar{\mathbb{G}}$ is the $d \times d$ diagonal matrix whose k th diagonal entry equals $\bar{w}_{ij}$ if the k th arc in $\mathbb{G}$ is (j, i) and (j, i) is also an arc in $\bar{\mathbb{G}}$, or 0 if the k th arc in $\mathbb{G}$ is not in $\bar{\mathbb{G}}$.

With the above definitions, it is not hard to verify that the m update equations in (6) can be written as

$$x(t+1) = x(t) - \frac{1}{2} \bar{J}(t)C'\bar{W}(t)C\bar{J}'(t)x(t), \quad (7)$$

where $\bar{J}(t) = J(t) \otimes I_n$ with $J(t)$ being the spanning incidence matrix of $\mathbb{N}(t)$, and $\bar{W}(t) = W(t) \otimes I_n$ with $W(t)$ being the spanning weight matrix of $\mathbb{N}(t)$. It is worth noting that all $W(t)$ are nonnegative diagonal matrices with the same size. It is also worth emphasizing that the definitions of C , $J(t)$, and $W(t)$ are based on the same ordering of the arcs in $\mathbb{N}$, and the equality (7) is independent of the ordering.

To proceed, we need the following concept and result.

A square matrix M is called *paracontracting* with respect to a vector norm $\|\cdot\|$ if $\|Mx\| \leq \|x\|$ and the strict inequality holds whenever $Mx \neq x$.

It is easy to see that any symmetric matrix is paracontracting with respect to the 2-norm if all its eigenvalues lie in the interval $(-1, 1]$.

For a square matrix M , we define its fixed point set as

$$\mathcal{F}(M) = \{x : Mx = x\}.$$

Paracontracting matrices have the following properties.

Lemma 6: Suppose that a finite set of square matrices $\{M_1, M_2, \dots, M_p\}$ are paracontracting with respect to the same vector norm. Let $\sigma(1), \sigma(2), \dots$ be an infinite sequence of integers taking values in $\{1, 2, \dots, p\}$ and $\mathcal{I}$ be the set of all integers that appears infinitely often in the sequence. Then for any initial vector $z(0)$, the sequence of vectors generated by $z(t+1) = M_{\sigma(t)}z(t)$ has a limit $z^* \in \bigcap_{i \in \mathcal{I}} \mathcal{F}(M_i)$.

The lemma is a special case of Theorem 1 in [19].

We also need the following lemmas.

Lemma 7: Let $\bar{W} = W \otimes I$, where W is a positive diagonal matrix. If $\bar{\mathbb{N}}$ is well-configured, then $\bar{J}C'\bar{W}C\bar{J}'$ has exactly n eigenvalues at zero, and all the remaining eigenvalues are positive.

Lemma 8: Let $\mathbb{G}$ be a symmetric, spanning subgraph of $\mathbb{N}$, W be the spanning weight matrix of $\mathbb{G}$, and J be the spanning incidence matrix of $\mathbb{G}$. Then all the eigenvalues of $I - \frac{1}{2}\bar{J}C'\bar{W}C\bar{J}'$ lie in $(-1, 1]$. If furthermore $\mathbb{G} = \mathbb{N}$, $I - \frac{1}{2}\bar{J}C'\bar{W}C\bar{J}'$ has exactly n eigenvalues at one and all the remaining eigenvalues lie in $(-1, 1)$.

The above lemma implies that each update matrix $(I - \frac{1}{2}\bar{J}(t)C'\bar{W}(t)C\bar{J}'(t))$ in (7) is paracontracting with respect to the 2-norm.

Lemma 9: Let $\mathbb{G}_1, \mathbb{G}_2, \dots, \mathbb{G}_p$ be a finite set of symmetric, spanning subgraphs of $\mathbb{G}$. If the union of $\mathbb{G}_1, \mathbb{G}_2, \dots, \mathbb{G}_p$ is $\mathbb{G}$, then $\text{kernel } C\bar{J}' = \text{kernel } C(\sum_{i=1}^p \bar{W}_i^{1/2} \bar{J}'_i)$, where J is the incidence matrix of $\mathbb{G}$, J_i is the spanning incidence matrix of $\mathbb{G}_i$, and W_i is the spanning weight matrix of $\mathbb{G}_i$.

V. ALGORITHM FOR DIRECTED CYCLES

Consider an m -vertex directed cycle $1 \rightarrow 2 \rightarrow \dots \rightarrow m \rightarrow 1$ whose local agreement equations are given in (3). The agents update their states as follows:

$$x_i(t+1) = x_i(t) - \frac{1}{2}P_i(x_i(t) - x_{i-1}(t)), \quad i \in \mathbf{m}, \quad (8)$$

where $P_i = C'_i(C_i C'_i)^{-1}C_i$ is a projection on $\mathcal{K}_i^\perp$.

Theorem 7: If $\mathbb{N}$ is an m -vertex directed cycle and $\bar{\mathbb{N}}$ is well-configured, then algorithm (8) will lead all m agents to reach a consensus exponentially fast for any initial states.

VI. CONCLUSION

In this paper, we have studied the problem of achieving a consensus in the face of limited information transfer between agents, in which each agent receives a linear function of the state of each of its neighbors; in the case when the linear function is realized by a matrix whose kernel is nonzero, the neighbor's state cannot be determined by the information transferred. From this perspective, the problem studied here is related to so-called privacy preserving consensus problems [20], which typically rely on carefully designed additive noise. The limited information idea here can be used to protect the privacy of agents' states without adding noise. The problem is also related to the compressed communication techniques which have been recently used to address the communication bottleneck in distributed optimization and machine learning [21].

The feasibility of the problem of interest has been termed as well-configuration. Sufficient conditions for a multi-agent system to be well-configured have been provided for different types of directed graphs. For well-configured multi-agent systems, provably correct distributed algorithms have been developed for a number of special cases of the problem. It

turns out that the state forms of these algorithms share similarity with so-called matrix-weighted consensus processes [22], [23]. Our results imply that the existing sufficient conditions for matrix-weighted consensus, which usually require a tree whose matrix-valued weights are all positive definite, can be significantly relaxed.

REFERENCES

- [1] J. Zhu, Y. Lin, J. Liu, and A. S. Morse. Reaching a consensus with limited information. 2022. arXiv:2204.00199 [math.OC].
- [2] M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *Journal of the ACM*, 27(2):228–234, 1980.
- [3] S. Mou, J. Liu, and A. S. Morse. A distributed algorithm for solving a linear algebraic equation. *IEEE Transactions on Automatic Control*, 60(11):2863–2878, 2015.
- [4] A. Nedić and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.
- [5] R. Olfati-Saber. Distributed Kalman filtering for sensor networks. In *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 5492–5498, 2007.
- [6] G. Antonelli. Interconnected dynamic systems: An overview on distributed control. *IEEE Control Systems Magazine*, 33(1):76–88, 2013.
- [7] A. Kashyap, T. Başar, and R. Srikant. Quantized consensus. *Automatica*, 43(7):1192–1203, 2007.
- [8] M. Cao, A. S. Morse, and B. D. O. Anderson. Agreeing asynchronously. *IEEE Transactions on Automatic Control*, 53(8):1826–1838, 2008.
- [9] L. Moreau. Stability of multi-agent systems with time-dependent communication links. *IEEE Transactions on Automatic Control*, 50(2):169–182, 2005.
- [10] M. J. Fischer, N. A. Lynch, and M. S. Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM*, 32(2):347–382, 1985.
- [11] J. N. Tsikiklis. *Problems in Decentralized Decision Making and Computation*. Ph.D dissertation, Department of Electrical Engineering and Computer Science, M.I.T., 1984.
- [12] M. El Chamie, J. Liu, and T. Başar. Design and analysis of distributed averaging with quantized communication. *IEEE Transactions on Automatic Control*, 61(12):3870–3884, 2016.
- [13] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems and Control Letters*, 53(1):65–78, 2004.
- [14] J. Liu, S. Mou, A. S. Morse, B. D. O. Anderson, and C. Yu. Deterministic gossiping. *Proceedings of the IEEE*, 99(9):1505–1524, 2011.
- [15] C. Godsil and G. Royle. *Algebraic Graph Theory*. Springer, 2001.
- [16] J. Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms and Applications*. Springer Science & Business Media, 2008.
- [17] H. E. Robbins. A theorem on graphs, with an application to a problem of traffic control. *American Mathematical Monthly*, 46(5):281–283, 1939.
- [18] 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*, pages 63–70, 2005.
- [19] L. Elsner, I. Koltracht, and M. Neumann. On the convergence of asynchronous paracontractions with applications to tomographic reconstruction from incomplete data. *Linear Algebra and its Applications*, 130:65–82, 1990.
- [20] Y. Mo and R. M. Murray. Privacy preserving average consensus. *IEEE Transactions on Automatic Control*, 62(2):753–765, 2017.
- [21] A. Koloskova, S. U. Stich, and M. Jaggi. Decentralized stochastic optimization and gossip algorithms with compressed communication. In *Proceedings of the 36th International Conference on Machine Learning*, pages 3478–3487, 2019.
- [22] P. Barooh and J. P. Hespanha. Graph effective resistance and distributed control: Spectral properties and applications. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 3479–3485, 2006.
- [23] Q. V. Tran, M. H. Trinh, and H.-S. Ahn. Discrete-time matrix-weighted consensus. *IEEE Transactions on Control of Network Systems*, 8(4):1568–1578, 2021.