This article has been accepted for publication in IEEE Control Systems Letters. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2022.3185419

Distributed Mode Computation in Open Multi-Agent Systems

Zoreh Al Zahra Sanai Dashti[1], Gabriele Oliva[2*], *Senior Member, IEEE,* Carla Seatzu[1], *Senior Member, IEEE,* Andrea Gasparri[3], *Senior Member, IEEE,* and Mauro Franceschelli[1], *Member, IEEE*

***Abstract*— Allowing Multi-Agent Systems (MAS) to compute the mode of the agents' initial values (i.e., the value with largest cardinality) represents a highly valuable building block for the development of complex decision-making tasks, as it allows agents to identify the central tendency of data or to implement majority voting processes while considering categorical opinions for which average or median values might not be possible to compute. This is especially challenging in the context of Open Multi-Agent Systems (OMAS), where agents are free to join or leave the network, as in this case the outcome of the mode computation process may vary depending on the current participants to the network. In this paper, we propose a novel OMAS mode computation framework where agents select a value from a finite set of alternatives, and compute the mode via the execution in parallel of a novel average-preserving distributed consensus procedure for each of the different alternatives. We complement the paper with simulation results that numerically demonstrate the effectiveness of the proposed approach.**

***Index Terms*— Distributed Mode Computation; Distributed Majority Voting; Open Multi-Agent Systems; Distributed Consensus**

I. INTRODUCTION

BEING able to compute the *mode* of a set of values is a fundamental building block for decision-making, as it allows to identify the central tendency when considering categorical data such as movies, car models or brands of soaps, for which it is not possible to compute average or median values. Notably, the mode has a deep connection with *majority voting* [1], [2]; in fact, considering a scenario where a group of decision-makers each select one out of a finite set of possible choices, e.g., candidates running for a political election or startups being evaluated for funding, the mode essentially coincides with the most popular choice. Interestingly, the computation of the mode finds application also in *ensemble learning* [3], [4] and has been theorized to represent a major underlying function of the brain's neocortex [5], [6].

In this view, being able to compute the mode of the opinion of a set of agents in a distributed fashion would represent a quite useful feature, as it would allow agents to implement complex coordination, voting and interaction strategies that involve categorical values and would not be possible to handle via standard consensus approaches. Distributed mode computation is particularly challenging in the context of *Open Multi-Agent Systems* (OMAS), where agents are free to join or leave the network. Yet, being able to implement a mode computation or majority voting process in OMAS would be highly beneficial, as in this case the outcome of the voting process may vary depending on the current participants to the network. Moreover, such a feature would enable the implementation of sophisticated coordination strategies in scenarios such as wireless sensor networks, where nodes could deplete their batteries or be switched off to save power, or mobile robots, where agents could be temporarily off-reach during exploration tasks.

A. State of the Art

In the context of *Multi-Agent Systems* (MAS), several distributed mode computation or majority voting approaches have been developed. In [7], a distributed voting process for agents holding one among several possible values is mapped into a gossip protocol where agents each choose a vertex of a polyhedron and collectively compute the resulting center of mass, which is then used to choose the closest vertex. In [8], an approach is developed, where agents can initially choose between two values, and then update their choice based on the most frequent value among k randomly chosen neighbors. In [9], an approach is discussed where nodes each maintain a set of possible values. In detail, when a node randomly wakes up, it selects a random neighbor, and then the one with the value set of smallest cardinality updates its value as the union of the value sets, while the other updates its set as the intersection. In [10], distributed multi-choice voting algorithms are developed considering agents that can communicate merely by sending beep signals. Such algorithms are shown to converge with high probability. In [11] an algorithm is developed where, when an

Department of Electrical and Electronic Engineering, University of Cagliari, Piazza D'Armi, 09123, Cagliari, Italy.
[2] University Campus Bio-Medico of Rome, via A. del Portillo 21, 00128 Rome, Italy.
[3] University Roma Tre, via della Vasca Navale 79, 00146 Rome, Italy.
[*] corresponding author. Email: g.oliva@unicampus.it
This work was supported in part by the Fondazione di Sardegna with grant "Formal Methods and Technologies for the Future of Energy Systems", n. F72F20000350007."
This work was supported by POR FESR Lazio Region Project RESIM under grant n. 228 A0375-2020-36673 (CUP: F89J21004860008).
This work has been supported by the European Commission under the grant agreement number 101016906 – Project CANOPIES.

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: Universita di Cagliari. Downloaded on June 27,2022 at 15:10:17 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Control Systems Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2022.3185419

agent wakes up, it computes the cardinality of the different values of its neighbors. Then, the following step is executed: decrease by one all cardinalities that are not zero and transmit such values to a neighbor that has not yet received messages since the node awoke. The awaken node continues until only one outcome remains, which becomes the new value for the node, terminating the step. Then, a new agent wakes up.

Notice that, in the literature, a fair amount of distributed consensus approaches for OMAS scenarios have been developed [12]–[21]. In particular, in [12], an algorithm robust to agents joining or leaving the network has been developed, although no stability analysis has been formally carried out. In [13] an algorithm has been developed under the assumption that agents' departure and arrival occurred at predetermined times, while in [14] it is assumed that each time an agent leaves the network, another one immediately joins it. The above approach has been extended in [15] to the case of time-varying network size. In [17], [18], agents estimate the time-varying average of a set of reference signals. In [19], stochastic consensus for OMAS based on a Bernoulli process was considered. In [20], a consensus algorithm for OMAS has been developed where agents track the median of time-varying reference signals. In [21] the interactions of agents over randomly induced discretized Laplacians is investigated.

B. Contribution

As discussed above, to the best of our knowledge, no distributed mode computation algorithm has been developed in the context of OMAS: in this paper we aim to fill this gap. , we consider an OMAS scenario and we develop a distributed algorithm that allows agents currently participating in the network to compute the mode of their voting preferences. The main building block of the proposed approach is a novel average-preserving consensus process that is suitable for OMAS, in that it allows to compute the average of the initial conditions of agents that are currently participating in the network. In this view, agents first implement multiple instances of the average-preserving consensus process (i.e., one for each discrete value), and then compute the mode by locally selecting the value associated with the largest among the scalar consensus outcomes. Notice that, in this paper, we assume that the total number of agents is upperbounded by a finite value.

II. Preliminaries

A. Notation

We denote vectors by boldface lowercase letters, matrices with plain uppercase letters and sets by uppercase italicized letters. We refer to the (i,j)-th entry of a matrix A by A_{ij}. We denote the $n\times n$ identity matrix by I_n and we use $O_{n\times n}$ to denote the $n\times m$ matrix with all zero entries. Moreover, we use $|\mathcal{S}|$ to denote the cardinality of a set $\mathcal{S}$ and we refer to the absolute value of a scalar $a\in\mathbb{R}$ by $\mathtt{abs}(a)$. A matrix A is said to be *nonnegative* if all its entries satisfy $A_{ij}\geq 0$. An $n\times m$ nonnegative matrix is called *row-stochastic* if $\sum_{j=1}^m A_{ij}=1$, and *column-stochastic* if $\sum_{j=1}^n A_{ji}=1$. A nonnegative matrix that is both row- and column-stochastic is referred to as *doubly stochastic*. Given an $n\times n$ matrix Q we define

$$\rho_2(Q)=\mathtt{abs}(\lambda_2(Q)), \tag{1}$$

where $\lambda_2(Q)$ is the eigenvalue of Q with second largest magnitude.

B. Network Topology in Open Multi-Agent Systems

In this paper we consider a scenario where agents interact on a discrete-time basis over a *time-varying graph*, where time variance is given by the fact that agents are free to leave or join the network. In particular, we assume that nodes join the network by creating links in an arbitrary way with nodes that are already present in the network. Notably, as it will be made clear later, nodes each have an initially chosen value, and we assume that nodes joining the network reset their state to the value they chose at the beginning even if they participated to the network at previous time instants. Moreover, once created, the links do not change unless one of the endpoint agents is disconnected. In case an agent is disconnected, all its incident links are removed. Notice that, in the proposed scenario, a disconnection is not abrupt; in particular, we assume that upon disconnection of an agent i, all its neighbors are aware of the disconnection and undertake appropriate actions as discussed later in the paper. In the following, we consider a situation where the overall number of agents is limited to $n_{\max}<\infty$ and each agent has a unique identifier in $\{1,\dots,n_{\max}\}$.

In this view, at each time instant t, the time-varying graph is given by $\mathcal{G}(t)=\{\mathcal{V}(t),\mathcal{E}(t)\}$, where $\mathcal{V}(t)\subseteq\{1,\dots,n_{\max}\}$ denotes the set of the indices of agents that are participating in the network at time t, while $E(t)\subseteq\{\mathcal{V}(t)\times\mathcal{V}(t)\}$ is the set of edges representing information exchange between agents; in other words, two agents i,j interact at time t if and only if $(i,j)\in\mathcal{E}(t)$. Notice that, in this paper, we assume $\mathcal{G}(t)$ is *undirected* at all time instants t, i.e., $(i,j)\in\mathcal{E}(t)$ if and only if $(j,i)\in\mathcal{E}(t)$. Moreover, we assume $\mathcal{G}(t)$ is connected[1] at all time instants t, i.e., each node in $\mathcal{V}(t)$ can be reached by each other node via a path composed of edges in $\mathcal{E}(t)$. Let us define the *neighborhood* $\mathcal{N}_i(t)$ of an agent i at time t as $\mathcal{N}_i(t)=\{j\in\mathcal{V}(t)\,|\,(i,j)\in\mathcal{E}(t)\}$. We denote by $\Delta_i(t)=|\mathcal{N}_i(t)|$ the *degree* of agent i at time t and by $|\mathcal{V}(t)|$ the number of agents participating in the network at time t. Moreover, for each time t, we denote the set $\mathcal{R}(t)$ of agents remaining in the network, the set $\mathcal{A}(t)$ of agents joining the network and the set $\mathcal{D}(t)$ of agents leaving the network, respectively, as $\mathcal{R}(t)=\mathcal{V}(t)\cap\mathcal{V}(t+1)$, $\mathcal{A}(t)=\mathcal{V}(t+1)\setminus\mathcal{V}(t)$ and $\mathcal{D}(t)=\mathcal{V}(t)\setminus\mathcal{V}(t+1)$. Notably, by definition, we have that $\mathcal{V}(t+1)=\mathcal{R}(t)\cup\mathcal{A}(t)$, $\mathcal{V}(t)=\mathcal{R}(t)\cup\mathcal{D}(t)$ and $\mathcal{V}(0)=\mathcal{A}(-1)$.

III. Problem Statement

Let us consider an open multi-agent system where agents interact over an undirected time-varying graph $\mathcal{G}(t)=\{\mathcal{V}(t),\mathcal{E}(t)\}$, as described in Section II-B. In particular, we consider a scenario where each agent is provided with

[1]The results discussed in this paper easily generalize to the case where the graph can be composed of multiple connected components.

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: Universita di Cagliari. Downloaded on June 27,2022 at 15:10:17 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Control Systems Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2022.3185419

a value h_i selected from a finite set $\mathcal{H}$ of possible values, with $|\mathcal{H}| = b \geq 2$. For simplicity, and without loss of generality, we hereby assume that the values are consecutive integers, i.e., $\mathcal{H} = \{1, \ldots, b\}$. For each time instant t, let us use $\boldsymbol{h}(t) \in \mathbb{R}^{|\mathcal{V}(t)|}$ to denote the stack of the values h_i for the agents $i \in \mathcal{V}(t)$ and let us use $\gamma_j(t)$ to denote the cardinality of the set of agents in $\mathcal{V}(t)$ holding the value j. Moreover, let us define the *mode* of $\boldsymbol{h}(t)$ as

$$m(t) = \arg \max_{j=1,\ldots,b} \{\gamma_j(t)\};$$

in other words, $m(t)$ is the value that appears most often in $\boldsymbol{h}(t)$ (or the subset of elements of $\mathcal{H}$, when multiple distinct values equally occur most often). Notice that $\boldsymbol{h}(t)$ is a function of time due to the agents joining and leaving the network at each instant of time. Our objective is to design a discrete-time distributed interaction strategy that allows agents to compute $m(t)$ by canceling out the influence of agents leaving the network.

A. Proposed Strategy

Let us first assume that $\mathcal{G}(t)$ is fixed, i.e., $\mathcal{G}(t) = \mathcal{G}(0)$ for all time instants t. In this case, when $b = 2$, if the agents execute an average consensus protocol with initial conditions

$$x_i(0) = \begin{cases} 1 & \text{if } h_i(0) = 1; \\ 0 & \text{otherwise,} \end{cases}$$

then the average consensus protocol yields a consensus value in the form

$$\widehat{x} = \frac{\gamma_1(0)}{|\mathcal{V}(0)|},$$

and we observe that, without the need to know $|\mathcal{V}(0)|$, the agents are able to compute the mode. In fact, if $\widehat{x} > 0.5$ then $\gamma_1(0) > 0.5|\mathcal{V}(0)|$ and thus the mode is equal to one; similarly, when $\widehat{x} < 0.5$ then $\gamma_1(0) < 0.5|\mathcal{V}(0)|$ and the mode is equal to two (when $\widehat{x} = 0.5$ the values one and two are equally distributed).

Let us generalize the above intuition to the case where b could be larger than 2. In this case, under the assumption that each agent knows b (this is not restrictive in practice, as we assumed the agents must be able to choose one of the b possible values), let us consider a scenario where each agent is provided with a *vectorial* initial condition

$$\boldsymbol{x}_i(0) = \left[x_i^{(1)}(0), \ldots, x_i^{(b)}(0)\right]^T \in \mathbb{R}^b,$$

where for all $i \in \{1, \ldots, n_{\max}\}$ and for all $\ell \in \{1, \ldots, b\}$ it holds

$$x_i^{(\ell)}(0) = \begin{cases} 1 & \text{if } h_i(0) = \ell; \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Assume the agents execute a vectorial average consensus protocol (or, equivalently, b scalar consensus protocols in parallel). Then, we have that the resulting vectorial average consensus value $\widehat{\boldsymbol{x}}$ is such that

$$\widehat{\boldsymbol{x}} = \frac{1}{|\mathcal{V}(0)|}\boldsymbol{\gamma}(0), \tag{3}$$

where $\boldsymbol{\gamma}(0) \in \mathbb{R}^b$ is the stack of the cardinalities $\gamma_i(0)$ for all values in $\mathcal{H}$. Therefore the agents, without the need to know $|\mathcal{V}(0)|$ are able to compute the mode by setting

$$m(t) = \arg \max_{i=1,\ldots,b} \{\gamma_i(0)\} = \arg \max_{i=1,\ldots,b} \{\widehat{x}_i\},$$

i.e., by choosing the index of the component of $\widehat{\boldsymbol{x}}$ with highest value.

At this point, let us consider an open multi-agent system where agents interact over an undirected time-varying graph $\mathcal{G}(t) = \{\mathcal{V}(t), \mathcal{E}(t)\}$, as described in Section II-B. In this case, given the setup of the proposed scheme, the usage of a classical consensus protocol would not suffice, as the arrival or departure of agents may significantly and permanently modify the consensus value, e.g., due to the past influence of agents leaving the network at some point. For this reason, we need a novel mechanism to track the consensus value: this is the objective of the next section. Notably, since we formulate the problem of computing the mode as a set of scalar average consensus problems in parallel, in the next section we consider the scalar consensus case.

IV. Average-Preserving Consensus for Open Multi-Agent Systems

In the previous section we show that, in order to compute the mode of their initial values $h_i(0)$, chosen from a set of b possible distinct values, the agents can execute b consensus procedures in parallel, where the initial condition $x_i^{(\ell)}(t)$ of the ℓ-th consensus procedure is chosen according to Eq. (2). In this section we develop an algorithm to let the agents execute the single consensus in an open multi-agent setting. For notational convenience, in the remainder of this section we use $x_i(t)$ to denote the state of the i-th agent in a scalar consensus procedure. Moreover, we use $\bar{x}_i$ to denote the initial state of the i-th agent. In particular, let us consider an open multi-agent system where agents interact over an undirected time-varying graph $\mathcal{G}(t) = \{\mathcal{V}(t), \mathcal{E}(t)\}$, as described in Section II-B.

Moreover, let us assume that each agent is provided with an initial piece of information $\bar{x}_i$ and let us define

$$c(t) = \frac{1}{|\mathcal{V}(t)|} \sum_{i \in \mathcal{V}(t)} \bar{x}_i, \tag{4}$$

i.e., $c(t)$ is the average of the initial condition of the agents that are participating in the network at time t; notably the time-varying number of agents $|\mathcal{V}(t)|$ is unknown to the agents and $c(t)$ varies as agents join and leave the network. In this section, our objective is to design a discrete-time distributed control protocol to let each agent track $c(t)$. As noted in the previous section, standard average consensus protocols may fail due to the effect of agents joining and leaving the network on the resulting consensus value. Moreover, we point out that any mechanism based on a re-initialization of a standard average consensus procedure each time an agent joins or leaves the network would require some mechanism to spread the need for re-initialization across the network, and thus any topological variation would be noticed only after some delay.

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: Universita di Cagliari. Downloaded on June 27,2022 at 15:10:17 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Control Systems Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2022.3185419

A. Proposed Protocol

Let $W(t)$ be an $n_{\max} \times n_{\max}$ matrix satisfying:

(i) $W_{ij}(t) > 0$ and $W_{ij}(t) = W_{ji}(t)$ whenever $(i,j) \in \mathcal{E}(t)$;
(ii) $W_{ij}(t) = 0$ for all $i \neq j$ such that $(i,j) \notin \mathcal{E}(t)$;
(iii) $W_{ii}(t) = 0$ for all $i \notin \mathcal{V}(t)$;
(iv) $I_{n_{\max}} + W(t)$ is nonnegative and doubly stochastic.

Moreover let us consider boolean variables $\phi_i(t), \psi_i(t)$ defined as follows:

$$\phi_i(t) = \begin{cases} 1, & \text{if } i \in \mathcal{R}(t); \\ 0, & \text{otherwise,} \end{cases} \quad \psi_i(t) = \begin{cases} 1, & \text{if } i \in \mathcal{A}(t); \\ 0, & \text{otherwise.} \end{cases}$$

Within the proposed protocol, at each time instant t each agent i that is participating in the network (either because it was already participating at time $t-1$ or because it joins the network at time t) maintains and updates an auxiliary variable $z_{ij}(t)$ for each of its neighbors in $\mathcal{N}_i(t)$. As discussed later, such a variable will represent the main ingredient to guarantee average-preservation in spite of the openness of the multi-agent system. In more detail, the protocol proposed for each agent $i \in \mathcal{V}(t+1)$ (i.e., considering only the agents that need to perform an update at time $t+1$) is as follows

$$x_i(t+1) = \psi_i(t)\overline{x}_i + \phi_i(t)x_i(t) + \phi_i(t) \underbrace{\left(\sum_{j \in \mathcal{N}_i(t) \cap \mathcal{R}(t)} W_{ij}(t)\big(x_j(t) - x_i(t)\big) + \sum_{j \in \mathcal{N}_i(t) \cap \mathcal{D}(t)} z_{ij}(t) \right)}_{u_i(t)}. \tag{5}$$

Moreover, each agent $i \in \mathcal{V}(t+1)$ maintains and updates auxiliary variable $z_{ij}(t)$ with the following dynamics

$$z_{ij}(t+1) = \phi_i(t)\phi_j(t)\left(z_{ij}(t) - W_{ij}(t)(x_j(t) - x_i(t))\right). \tag{6}$$

In other words, $z_{ij}(t+1)$ is zero whenever either $i \notin \mathcal{R}(t)$ or $j \notin \mathcal{R}(t)$ (or both). Notably, in the proposed approach, each agent must maintain $O(|\mathcal{N}_i(t)|b)$ variables at each time instant. In the worst-case of a complete graph where $|\mathcal{N}_i(t)| = n-1$, amounts to $O(nb)$ variables. Therefore, the proposed approach is particularly suitable when b is limited (i.e., when each agent has to choose in a small set of possible alternatives) and $|\mathcal{N}_i(t)| \ll n$, e.g., in the case of mobile robots or sensors with limited communication radius.

B. Convergence Analysis

In this subsection, we show that the sum of the initial conditions of agents participating in the network is preserved at all times and that, assuming the network becomes fixed at some time instant t^*, the state $x_i(t)$ of the agents in $\mathcal{V}(t^*)$ asymptotically reaches the average of the initial conditions of the agents in $\mathcal{V}(t^*)$.

Theorem 1: Let us consider an open multi-agent system where agents have dynamics described by Eqs. (5) and (6) and interact over an undirected time-varying graph $\mathcal{G}(t) = \{\mathcal{V}(t), \mathcal{E}(t)\}$, as described in Section II-B. Moreover, suppose $W(t)$ satisfies conditions (i)–(iv) for all time instants t. Then, for all $t \geq 0$, it holds

$$\sum_{i \in \mathcal{V}(t)} x_i(t) = \sum_{i \in \mathcal{V}(t)} \overline{x}_i. \tag{7}$$

Proof: In order to prove our statement, we observe that, by construction, it holds $\mathcal{V}(0) = \mathcal{A}(-1)$, therefore

$$\sum_{i \in \mathcal{V}(0)} x_i(0) = \sum_{i \in \mathcal{V}(0)} \overline{x}_i,$$

and thus our claim holds true for $t = 0$. Let us now prove that the claim holds true at time $t+1$ for all $t \geq 0$. To this end, by using Eqs. (5) and (6), we note that for all $i \in \mathcal{R}(t)$ it holds

$$\begin{aligned} x_i(t+1) + \sum_{j \in \mathcal{N}_i(t+1)} z_{ij}(t+1) &= x_i(t) + \sum_{j \in \mathcal{N}_i(t) \cap \mathcal{R}(t)} W_{ij}(x_i(t) - x_j(t)) \\ &+ \sum_{j \in \mathcal{N}_i(t) \cap \mathcal{D}(t)} z_{ij}(t) + \sum_{j \in \mathcal{N}_i(t) \cap \mathcal{R}(t)} z_{ij}(t) - \sum_{j \in \mathcal{N}_i(t) \cap \mathcal{R}(t)} W_{ij}(x_i(t) - x_j(t)) \\ &= x_i(t) + \sum_{j \in \mathcal{N}_i(t)} z_{ij}(t). \end{aligned}$$

Thus, by defining $t_i^\star$ as the last time agent $i \in \mathcal{R}(t)$ joined/activated before time $t+1$ (notably, by construction, such time $t^* \leq t$ exists for any $t \geq 0$), it holds

$$x_i(t+1) + \sum_{j \in \mathcal{N}_i(t+1)} z_{ij}(t+1) = x_i(t_i^\star) + \sum_{j \in \mathcal{N}_i(t_i^\star)} z_{ij}(t_i^\star).$$

At this point we observe that, by construction, all terms $z_{ij}(t_i^\star)$ are equal to zero and $x_i(t_i^\star) = \bar{x}_i$; therefore we have that, for all $i \in \mathcal{R}(t)$ it holds

$$x_i(t+1) + \sum_{j \in \mathcal{N}_i(t+1)} z_{ij}(t+1) = \bar{x}_i.$$

Moreover, for all $i \in \mathcal{A}(t)$ we have that, by definition $x_i(t+1) = \bar{x}_i$ and $z_{ij}(t+1) = 0$; hence, it holds

$$x_i(t+1) + \sum_{j \in \mathcal{N}_i(t+1)} z_{ij}(t+1) = \bar{x}_i$$

Therefore, since $\mathcal{V}(t+1) = \mathcal{A}(t) \cup \mathcal{R}(t)$ and $\mathcal{A}(t) \cap \mathcal{R}(t) = \emptyset$, we obtain

$$\sum_{i \in \mathcal{V}(t+1)} x_i(t+1) + \sum_{i \in \mathcal{V}(t+1)} \sum_{j \in \mathcal{N}_i(t+1)} z_{ij}(t+1) = \sum_{i \in \mathcal{V}(t+1)} \bar{x}_i.$$

The proof follows noting that, by construction $z_{ij}(t+1) = -z_{ji}(t+1)$ and thus

$$\sum_{i \in \mathcal{V}(t+1)} \sum_{j \in \mathcal{N}_i(t+1)} z_{ij}(t+1) = 0.$$

This completes our proof. ∎

We are now in position to prove that, assuming no agent leaves or joins the network after a time instant t^*, the agents' states converge to the average of the initial conditions of the nodes that are participating in the network at time t^*.

Theorem 2: Let us consider an open multi-agent system where agents have dynamics described by Eqs. (5) and (6) and interact over an undirected time-varying graph $\mathcal{G}(t) = \{\mathcal{V}(t), \mathcal{E}(t)\}$, as described in Section II-B. Moreover, suppose $W(t)$ satisfies conditions (i)–(iv) for all time instants t. Further to that, suppose that there

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: Universita di Cagliari. Downloaded on June 27,2022 at 15:10:17 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Control Systems Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2022.3185419

is a finite time instant t^* such that $\mathcal{G}(t) = \mathcal{G}(t^*)$ for all $t \geq t^*$, i.e, no agent joins or leaves the network from time t^* on. Then, the state $x_i(t)$ of the agents $i \in \mathcal{V}(t^*)$ asymptotically converges to $c(t^*)$, i.e., $\lim_{t\to\infty} |x_i(t) - c(t^*)| = 0$ for all $i \in \mathcal{V}(t^*)$; moreover, convergence is exponential, and the convergence rate is given by $\rho_2(Q(t^*)) \in (0,1)$, where $Q(t^*) = I_{|\mathcal{V}(t^*)|} + HW(t^*)H^T$, with $H = \begin{bmatrix} I_{|\mathcal{V}(t^*)|} & O_{|\mathcal{V}(t^*)|\times q} \end{bmatrix}$, and $q = n_{\max} - |\mathcal{V}(t^*)|$, while $\rho_2(\cdot)$ is defined in Eq. (1).

Proof: In order to prove our result, we observe that, assuming no agent joins or leaves the network from t^* on, for all $t \geq t^*$ it holds $c(t) = c(t^*)$, $\mathcal{R}(t) = \mathcal{R}(t^*)$, $\mathcal{A}(t) = \mathcal{D}(t) = \emptyset$, and thus $\mathcal{V}(t+1) = \mathcal{R}(t^*)$. Therefore, we have that, for all $i \in \mathcal{R}(t^*)$ and for all $t \geq t^*$ $\phi_i(t) = 1$ and, by Eq. (5),

$$
\begin{aligned}
e_i(t+1) &= \underbrace{x_i(t+1)}_{x_i(t)+u_i(t)} - \underbrace{c(t+1)}_{c(t)} = e_i(t) + u_i(t) \\
&= e_i(t) + \sum_{j\in\mathcal{N}_i(t)\cap\mathcal{R}(t)} W_{ij}(t)\big(\underbrace{x_j(t) - x_i(t)}_{e_j(t)-e_i(t)}\big) - \underbrace{\sum_{j\in\mathcal{N}_i(t)\cap\mathcal{D}(t)} z_{ij}(t)}_{0} \\
&= e_i(t) + \sum_{j\in\mathcal{N}_i(t)\cap\mathcal{R}(t)} W_{ij}(t)\big(e_j(t) - e_i(t)\big),
\end{aligned}
\tag{8}
$$

where $u_i(t)$ is defined in Eq. (5) and the last equation holds since, for all $t \geq t^*$, we have that $\mathcal{D}(t) = \emptyset$, and thus $\mathcal{N}_i(t) \cap \mathcal{R}(t) = \mathcal{N}_i(t)$. Stacking the above equation for all agents, we obtain $\boldsymbol{e}(t+1) = P(t)\boldsymbol{e}(t)$, where $P(t) = P(t^*) = I_{n_{\max}} + W(t^*)$. For the sake of simplicity, and without loss of generality, let us assume that the identifiers of the nodes in $\mathcal{V}(t^*)$ correspond to the first $|\mathcal{V}(t^*)|$ ones and let us decompose $\boldsymbol{e}^T(t)$ as

$$
\boldsymbol{e}^T(t) = \begin{bmatrix} \boldsymbol{\epsilon}^T(t) & \boldsymbol{\eta}^T(t) \end{bmatrix}^T,
$$

where $\boldsymbol{\epsilon}(t)$ collects the disagreement of the first $|\mathcal{V}(t^*)|$ agents (i.e., the ones participating in the network from time t^* on) and $\boldsymbol{\eta}(t)$ collects the disagreement of the remaining agents. Considering only the agents in $\mathcal{V}(t^*)$, we have that

$$
\boldsymbol{\epsilon}(t+1) = Q(t^*)\boldsymbol{\epsilon}(t). \tag{9}
$$

By construction, $Q(t^*)$ is nonnegative, doubly stochastic, and the graph induced by considering its positive off-diagonal entries is connected. Therefore, Eq. (9) is essentially a standard discrete-time average consensus process, which is known [22] to exponentially converge to the average of the initial conditions (in our case, the average of the agents' states at time t^*) with convergence rate equal to $\rho_2(Q(t^*)) \in (0,1)$. To conclude we point out that, since we established in Theorem 1 that the sum of the initial conditions is preserved, we have that, by construction, it holds

$$
\begin{aligned}
\frac{1}{|\mathcal{V}(t^*)|}\sum_{i\in\mathcal{V}(t^*)} \epsilon_i(t^*) &= \frac{1}{|\mathcal{V}(t^*)|}\sum_{i\in\mathcal{V}(t^*)} (x_i(t) - c(t^*)) \\
&= \frac{1}{|\mathcal{V}(t^*)|}\sum_{i\in\mathcal{V}(t^*)} \bar{x}_i - c(t^*) = 0;
\end{aligned}
$$

hence $\boldsymbol{\epsilon}(t)$ approaches zero as t approaches infinity. This completes our proof. ■

Remark 1: A simple, yet effective choice for the weights $W_{ij}(t)$ that satisfies points (i)–(iv) is represented by the so-called *Metropolis* weights [23], defined as

$$
W_{ij}(t) = \begin{cases} \dfrac{1}{1+\max\{\Delta_i(t), \Delta_j(t)\}} & (i,j) \in \mathcal{E}(t), \\ -\displaystyle\sum_{j\in\mathcal{N}_i(t)} W_{ij}(t) & i = j, \\ 0 & \text{otherwise}. \end{cases}
\tag{10}
$$

In fact, with this choice, $W(t)$ is by construction symmetric, nonnegative, doubly stochastic and has a structure that corresponds to the graph $\mathcal{G}(t)$.

Remark 2: The results in Theorems 1–2 easily generalize to the case where $G(t)$ has multiple connected components. In fact, using the same argument as in Theorem 1, it can be shown that the sum of the states of the nodes currently in a component is equal to the sum of their initial values. Moreover, with the same logic as in Theorem 2, if from some time t^* the topology becomes fixed then each connected component reaches the average of the initial values of the agents in that component, with exponential convergence rate.

V. Simulations

Let us consider an open multi-agent system featuring $n_{\max} = 30$ agents, each provided with a value $h_i(0)$ chosen among four possible alternatives, i.e., $b = 4$. As discussed in Section III-A, the mode is computed by executing four scalar consensus procedures in parallel, where the initial condition $x_i^{(\ell)}(0)$ of the i-th agent in the ℓ-th scalar consensus procedure is chosen according to Eq. (2). We assume that the agents are initially all participating in the network (i.e., $|\mathcal{V}(0)| = n_{\max}$) and that the graph $G(0)$ modeling their initial interaction is an undirected and connected Erdös-Rényi graph [24] with link formation probability $p = 0.35$ (not reported here due to page-limit restrictions). Let us consider a scenario where, at time $t = 30$, a subset of 15 agents leaves the network, while at time $t = 61$, ten agents that left the network join it again. In all cases, the graph $\mathcal{G}(t)$ remains connected. Table I reports the distribution over time of the cardinalities $\gamma_j(t)$ for the different values $j \in \mathcal{H}$ chosen by the agents, along with the mode $m(t)$. Notably, for $t \in [30, 60]$ the mode changes from $m(t) = 4$ to $m(t) = 1$, since seven agents holding the value $h_i = 4$ and six agents holding $h_i = 3$ leave the network, but only one agent holding $h_i = 1$ does. At time $t = 61$, six agents holding $h_i = 3$ but only two agents with $h_i = 4$ join again the network and the mode becomes $m(t) = 3$.

	$\gamma_1(t)$	$\gamma_2(t)$	$\gamma_3(t)$	$\gamma_4(t)$	$m(t)$
$t \in [0, 29]$	6	4	9	11	4
$t \in [30, 60]$	5	3	3	4	1
$t \geq 61$	6	4	9	6	3

TABLE I**:** Cardinalities $\gamma_j(t)$ and mode $m(t)$ over time.

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: Universita di Cagliari. Downloaded on June 27,2022 at 15:10:17 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Control Systems Letters. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2022.3185419

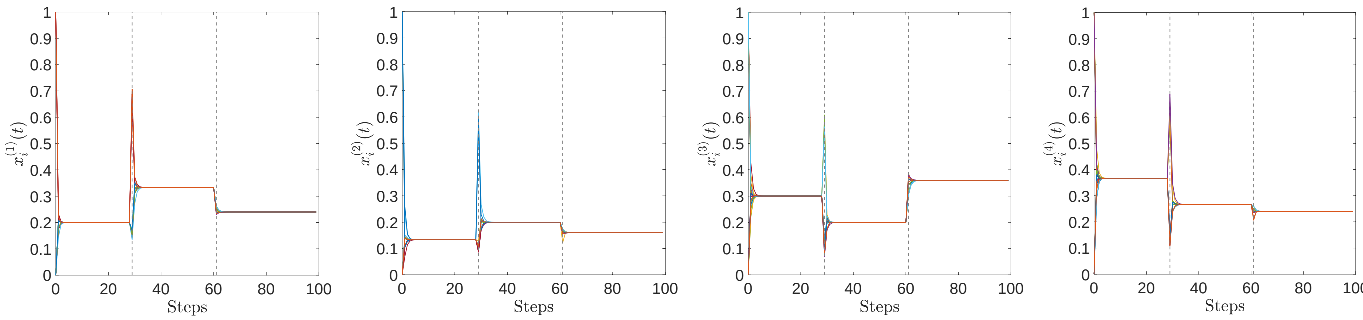

Fig. 1**:** Evolution of the proposed algorithm for tracking the mode in open multi-agent system, which amounts to four instances of the scalar consensus procedure proposed in Section IV. Within each procedure, the agents are able to compute a component of $\widehat{x}$ (Eq. (3)), and the mode $m(t)$ is chosen as the index of the component of $\widehat{x}$ with the largest magnitude. The dotted vertical lines correspond to the time instants $t = 30$ and $t = 61$, respectively, where some of the agents leave and/or join the network.

Figure 1 shows the evolution of four scalar average consensus procedures for open multi-agent systems, developed in Section IV, which are run in parallel and allow the agents to compute the four components of the vector $\widehat{\boldsymbol{x}}$, defined in Eq. (3). According to the figure, when agents join or leave the network, a new transient is started and, within each scalar procedure, the average of the initial conditions of agents participating in the network is asymptotically computed. This allows the agents to compute the mode $m(t)$ by selecting the index of the consensus variable of largest magnitude. In particular, we observe that, indeed, the agents are able to compute $m(t)$ as being equal to four for $t \in [0, 29]$, to one for $t \in [30, 60]$ and, again, to four for $t \geq 61$.

VI. Conclusions

In this paper, a distributed strategy to allow agents in an OMAS to track the mode of the initial conditions of those agents that are currently participating in the network has been developed, which is thus essentially a distributed majority voting process. The approach is based on a novel average-preserving distributed consensus protocol, which allows agents participating in the network to compute the average of their initial conditions by canceling out the influence of agents leaving the network. Based on this, we develop a strategy to let agents track the *mode* of a discrete and finite set of values. In particular, our strategy is based on the execution in parallel of the aforementioned average-preserving distributed consensus. Future work will be devoted to extending the approach to guarantee finite-time tracking and use the majority voting process as a way to combine the outcome of classifiers trained at each agent.

References

[1] R. Rose and H. Mossawir, "Voting and elections: A functional analysis," *Political studies*, vol. 15, no. 2, pp. 173–201, 1967.
[2] J. Cai, J. L. Garner, and R. A. Walkling, "Electing directors," *The Journal of Finance*, vol. 64, no. 5, pp. 2389–2421, 2009.
[3] Y. Freund, R. E. Schapire *et al.*, "Experiments with a new boosting algorithm," in *icml*, vol. 96. Citeseer, 1996, pp. 148–156.
[4] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
[5] J. Hawkins, S. Ahmad, and Y. Cui, "A theory of how columns in the neocortex enable learning the structure of the world," *Frontiers in neural circuits*, p. 81, 2017.
[6] J. Hawkins, *A thousand brains: A new theory of intelligence*. Basic Books, 2021.
[7] F. Bénézit, P. Thiran, and M. Vetterli, "The distributed multiple voting problem," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 791–804, 2011.
[8] M. A. Abdullah and M. Draief, "Global majority consensus by local majority polling on graphs of a given degree sequence," *Discrete Applied Mathematics*, vol. 180, pp. 1–10, 2015.
[9] S. Salehkaleybar, A. Sharif-Nassab, and S. J. Golestani, "Distributed voting/ranking with optimal number of states per node," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 1, no. 4, pp. 259–267, 2015.
[10] B. Ghojogh and S. Salehkaleybar, "Distributed voting in beep model," *Signal Processing*, vol. 177, p. 107732, 2020.
[11] H. Bandealinaeini and S. Salehkaleybar, "Broadcast distributed voting algorithm in population protocols," *IET Signal Processing*, vol. 14, no. 10, pp. 846–853, 2021.
[12] M. Zhu and S. Martínez, "Discrete-time dynamic average consensus," *Automatica*, vol. 46, no. 2, pp. 322–329, 2010.
[13] J. M. Hendrickx and S. Martin, "Open multi-agent systems: Gossiping with deterministic arrivals and departures," in *54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Sept 2016, pp. 1094–1101.
[14] ——, "Open multi-agent systems: Gossiping with random arrivals and departures," in *57th IEEE Conference on Decision and Control (CDC)*, Dec 2017.
[15] C. Monnoyer de Galland, S. Martin and J. M. Hendrickx, "Modelling Gossip Interactions in Open Multi-Agent Systems", arXiv, 2009.02970v2, 2022.
[16] M. Abdelrahim, J. M. Hendrickx, and W. P. M. H. Heemels, "Max-consensus in open multi-agent systems with gossip interactions," in *57th IEEE Conference on Decision and Control (CDC)*, Dec 2017.
[17] M. Franceschelli and P. Frasca, "Stability of open multi-agent systems and applications to dynamic consensus," *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2326 - 2331, 2021.
[18] M. Franceschelli and P. Frasca, "Proportional dynamic consensus in open multi-agent systems," in *58th IEEE Conference on Decision and Control (CDC)*, Dec. 2018.
[19] V. S. Varma, I. C. Morarescu, and D. Nesic, "Open multi-agent systems with discrete states and stochastic interactions," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 375–380, July 2018.
[20] Z. A. Z. Sanai Dashti, C. Seatzu, and M. Franceschelli, "Dynamic consensus on the median value in open multi-agent systems," in *58th IEEE Conference on Decision and Control (CDC)*, Dec 2019.
[21] R. Vizuete, P. Frasca, and E. Panteley, "On the influence of noise in randomized consensus algorithms," IEEE Control Systems Letters, vol. 5, no. 3, pp. 1025–1030, 2020.
[22] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.
[23] F. Bullo, *Lectures on Network Systems*. Kindle Direct Publishing, 2019.
[24] P. Erdős, A. Rényi *et al.*, "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci*, vol. 5, no. 1, pp. 17–60, 1960.

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: Universita di Cagliari. Downloaded on June 27,2022 at 15:10:17 UTC from IEEE Xplore. Restrictions apply.