Cluster Assignment in Multi-Agent Systems

Miel Sharf
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology
Stockholm, Sweden
sharf@kth.se

Daniel Zelazo
Faculty of Aerospace Engineering
Technion - Israel Institute of Technology
Haifa, Israel
dzelazo@technion.ac.il

Abstract—**We study cluster assignment in multi-agent networks. We consider homogeneous diffusive networks, and focus on design of the graph that ensures the system will converge to a prescribed cluster configuration, i.e., specifying the number of clusters and agents within each cluster. Leveraging recent results from cluster synthesis, we show that it is possible to design an oriented graph such that the action of the automorphism group of the graph has orbits of predetermined sizes, guaranteeing that the network will converge to the prescribed cluster configuration. We provide upper and lower bounds on the number of edges that are needed to construct these graphs along with a constructive approach for generating these graphs. We support our analysis with some numerical examples.**

Index Terms—**Multi-Agent Systems, Graph Theory, Clustering, Diffusively-Coupled Systems**

I. INTRODUCTION

The process of reaching an agreement between agents is one of the fundamental tasks for a multi-agent system (MAS). Indeed, agreement protocols, the decision rule implemented by each agent that enables them to distributedly reach agreement, appear across many diverse fields. These include distributed computation [1], robotics [2], biochemical systems [3], and sensor networks [4]. A natural extension to the agreement problem is the *cluster agreement* problem, which seeks to drive agents into groups. All the agents within the same group should then reach an agreement. The clustering problem also appears in many areas including neuroscience [5], biomimicry of swarms [6], and social networks [7].

Various approaches have been used to study clustering, e.g. exploiting the structural balance of the underlying graph [8], pinning control [9], inter-cluster nonidentical inputs [10], and network optimizaton [11]. Our approach to the cluster agreement problem is to leverage notions of *symmetry* within a multi-agent system. Symmetry of graphs has recently emerged as an important property for multi-agent systems, in particular in the study of controllability and observability properties of these systems [12]–[15].

In our recent work [16], we introduced the notion of the *weak automorphism group of an MAS*. This new notion of symmetry for MAS combines two ideas: the automorphism group for graphs and weak equivalence of dynamical systems. Graph automorphisms are known to capture notions of symmetries for graphs, while weak equivalence of dynamical systems

aims to characterize similarities of heterogeneous dynamical agents in terms of their achievable steady-states. The weak automorphisms of an MAS can be thought of, therefore, as a permutation of the nodes in the underlying graph that preserves graph symmetries together with certain input-output properties (i.e., steady-state maps) of the associated agents. We focused on clustering for *diffusively coupled networks*, as shown in Figure 1. Under appropriate passivity assumptions for the agent dynamics, we showed that these diffusively coupled networks converge to a clustered solution in the steady-state, where two agents are in the same cluster if and only if there exists a weak automorphism mapping one to the other. Thus, the clustering of the MAS can be understood by studying the action of the weak automorphism group on the underlying interaction graph.

In this work we focus on homogeneous networks, that is networks where the agent dynamics are all identical, noting that the weak automorphism group of the network is identical to the automorphism group of the underlying graph in this case. The problem we aim to solve is how to design graphs that ensure the networked system will converge to a prescribed cluster configuration, i.e., specifying the number of clusters and the number of agents within each cluster. Employing tools from group theory, we show that it is possible to design an oriented graph such that the action of the automorphism group of the graph has orbits of specified sizes determined by the designer. We provide upper and lower bounds on the number of edges that are needed to construct these graphs. The proof of these results in turn provide a constructive approach for generating these graphs. We support our analysis with some numerical examples.

The rest of the paper is organized as follows. Section II reviews basic concepts related to network systems and group theory required to define a notion of symmetry for multi-agent systems. In Section III, the main results related to cluster assignment are given. This section also presents some numerical studies to demonstrate the theory. Finally, some concluding remarks are offered in Section IV.

Notations: This work employs basic notions from graph theory [17]. An undirected graph $\mathcal{G} = (\mathbb{V}, \mathbb{E})$ consists of a finite set of vertices $\mathbb{V}$ and edges $\mathbb{E} \subset \mathbb{V} \times \mathbb{V}$. We denote by $k = \{i, j\} \in \mathbb{E}$ the edge that has ends i and j in $\mathbb{V}$. For each edge k, we pick an arbitrary orientation and denote $k = (i, j)$ when $i \in \mathbb{V}$ is the *head* of edge k and $j \in \mathbb{V}$ the
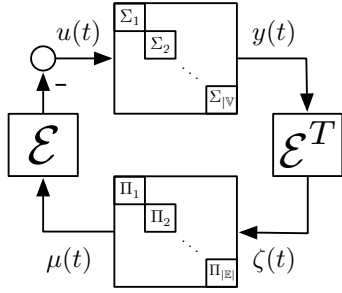
Fig. 1. A diffusively coupled network.

tail. A path is a sequence of distinct nodes $v_1, v_2, \ldots, v_n$ such that $\{v_i, v_{i+1}\} \in \mathbb{E}$ for all i. A cycle is the union of a path $v_1, \ldots, v_n$ with the edge $\{v_1, v_n\}$. A simple cycle is a cycle whose vertices are all distinct. A graph is called connected if there is a path between any two vertices, and a graph is called a tree if it is connected but contains no simple cycles.

The incidence matrix of $\mathcal{G}$, denoted $\mathcal{E} \in \mathbb{R}^{|\mathbb{E}| \times |\mathbb{V}|}$, is defined such that for edge $k = (i, j) \in \mathbb{E}$, $[\mathcal{E}]_{ik} = +1$, $[\mathcal{E}]_{jk} = -1$, and $[\mathcal{E}]_{\ell k} = 0$ for $\ell \neq i, j$. For a graph $\mathcal{G}$, an automorphism of $\mathcal{G}$ is a permutation $\psi : \mathbb{V} \to \mathbb{V}$ such that i is connected to j if and only if $\psi(i)$ is connected to $\psi(j)$. We denote its automorphism group by $\text{Aut}(\mathcal{G})$. Lastly, the greatest common divisor of two positive integers m, n is denoted by $\gcd(m, n)$, and their least common multiple is denoted by $\text{lcm}(m, n)$. Note that $\text{lcm}(m, n) = \frac{mn}{\gcd(m,n)}$ always holds.

II. Symmetries in Networked Systems

In this section we provide background on the notion of symmetries for multi-agent systems originally proposed in [16].

A. Diffusively Coupled Networks

In this subsection, we describe the structure of the network dynamical system studied in [18]. Consider a collection of agents interacting over a network $\mathcal{G} = (\mathbb{V}, \mathbb{E})$. The nodes $i \in \mathbb{V}$ are assigned dynamical systems Σ_i, and the edges $e \in \mathbb{E}$ are assigned controllers Π_e, having the following form:

$$\Sigma_i : \begin{cases} \dot{x}_i = f_i(x_i, u_i) \\ y_i = h_i(x_i, u_i), \end{cases} \quad \Pi_e : \begin{cases} \dot{\eta}_e = \phi_e(\eta_e, \zeta_e) \\ \mu_e = \psi_e(\eta_e, \zeta_e) \end{cases} . \quad (1)$$

We consider stacked vectors of the form $u = [u_1^T, ..., u_{|\mathbb{V}|}^T]^T$ and similarly for y, ζ and μ. The network system is diffusively coupled with the controller input described by $\zeta = \mathcal{E}^T y$, and the control input to each system by $u = -\mathcal{E}\mu$, where $\mathcal{E}$ is an incidence matrix of the graph $\mathcal{G}$. This structure is denoted by the triplet $(\mathcal{G}, \Sigma, \Pi)$, and is illustrated in Fig. 1. In this work we focus on homogeneous networks, i.e., where all the agent dynamics and control dynamics are the same. In [18] it was shown that the network converges to a steady-state satisfying the interconnection constraints if the agents and controllers are (output-strictly) maximum equilibrium independent passive (MEIP). This property can be thought of as an extension of equilibrium independent passivity developed in [19]. The details of these definitions are not essential for the development of this work, and the interested reader is referred

to [18], [20] for more details. For the rest of this paper, we will assume one of the following two alternatives. If this is not the case, see [21], [22] for plant augmentation techniques.

Assumption 1. *The agents Σ_i are output-striclty MEIP and the controllers Π_e are MEIP.*

Assumption 2. *The agents Σ_i are MEIP and the controllers Π_e are output-strictly MEIP.*

A final technical definition is needed to characterize the steady-states of the network. Indeed, embedded in our passivity assumption is a requirement that each agent and controller converge to steady-state outputs given a constant input. This allows us to define a relation between constant inputs to constant outputs that we call the *steady-state input/output relation* of a system; see [18]. We denote the steady-state input-output relations of the node i and the edge e by k_i and γ_e, respectively. For example, for agent i, we say that $(\mathrm{u}_i, \mathrm{y}_i)$ is a steady-state input/output pair if $\mathrm{y}_i \in k_i(\mathrm{u})$.

Definition 1 ([16]). *Two dynamical systems Σ_1 and Σ_j are called* weakly equivalent *if their steady-state input-output relations are identical.*

Examples of weakly equivalent systems are given in [16].

B. Group Theory, Graph Automorphisms, and Symmetric MAS

The main tool we use to facilitate clustering is symmetry, which is usually modelled using the mathematical theory of group theory [23]. There are many possible ways to define what a group is, and we choose the most concrete one:

Definition 2. *Let X be a set, and let $\mathbb{G}$ be a collection of invertible functions $X \to X$. Then $\mathbb{G}$ is called a group if for any $\mathbb{G} \ni f, g : X \to X$, both the composite function $f \circ g$ and the inverse function f^{-1} belong to $\mathbb{G}$.*

Colloquially, the group $\mathbb{G}$ defines a collection of symmetries of the set X. The action of the group $\mathbb{G}$ on X allows us to identify certain elements of X which are symmetric.

Definition 3. *Let $\mathbb{G}$ be a group of functions $X \to X$. We say that $x, y \in X$ are* exchangeable *(under the action of $\mathbb{G}$) if there is some $f \in \mathbb{G}$ such that $f(x) = y$. The* orbit *of $x \in X$ is the set of elements which are exchangeable with X.*

Exchangeability was considered in [16] to identify the clustering behavior of a multi-agent system. Namely, the different clusters corresponded to the different orbits of a certain group.

Proposition 1. *Let $\mathbb{G}$ be a group of functions $X \to X$. Then the set X can be written as the union of disjoint orbits.*

In this paper, we focus on the automorphism group $\text{Aut}(\mathcal{G})$ of an (oriented) graph $\mathcal{G}$. Namely, fixing an oriented graph $\mathcal{G} = (\mathbb{V}, \mathbb{E})$, a graph automoprhism is a permutation $\psi : \mathbb{V} \to \mathbb{V}$ such that if $(i, j) \in \mathbb{E}$, then $(\psi(i), \psi(j)) \in \mathbb{E}$. In that case, we abuse the notation and say that $\text{Aut}(\mathcal{G})$ acts on $\mathcal{G}$ (rather than on $\mathbb{V}$).

Finally, we combine the notions of automorphisms for graphs with diffusively coupled networks comprised of agents that are weakly equivalent.

Definition 4. *Let $(\mathcal{G}, \Sigma, \Pi)$ be any multi-agent system for SISO agents. A* weak automorphism *of a MAS is a map $\psi : \mathbb{V} \to \mathbb{V}$ such that the following conditions hold:*

i) *The map ψ is an automorphism of the graph $\mathcal{G}$.*
ii) *For any $i \in \mathbb{V}$, Σ_i and $\Sigma_{\psi(i)}$ are weakly equivalent.*
iii) *For any $e \in \mathbb{E}$, Π_e and $\Pi_{\psi(e)}$ are weakly equivalent.*
iv) *The map ψ preserves edge orientation.*

We denote the collection of all weak automorphisms of $(\mathcal{G}, \Sigma, \Pi)$ by $\mathrm{Aut}(\mathcal{G}, \Sigma, \Pi)$. Naturally, this is a subgroup of the group of automorphisms $\mathrm{Aut}(\mathcal{G})$ of the graph $\mathcal{G}$.

III. CLUSTER ASSIGNMENT IN MAS

We now turn our attention to the problem of clustering in MAS. Specifically, we focus on the case where the agents are homogeneous, i.e. they have the exact same model. In that case, we restrict ourselves by requiring that the edge controllers (1) are also homogeneous.

In [16] we proposed a symmetry-based framework to understand the clustering behaviour of a multi-agent system, through Definition 4. The main result from [16] can be summarized below.

Theorem 1 ([16])**.** *Consider the diffusively-coupled system $(\mathcal{G}, \Sigma, \Pi)$, and suppose that either Assumption 1 or Assumption 2 hold. Then for any steady-state $\mathrm{y} = \begin{bmatrix} \mathrm{y}_1 & \cdots & \mathrm{y}_{|\mathbb{V}|} \end{bmatrix}^T$ of the closed-loop and any weak automorphism $\psi \in \mathrm{Aut}(\mathcal{G}, \Sigma, \Pi)$, it follows that $P_\psi \mathrm{y} = \mathrm{y}$, where P_ψ is the permutation matrix representation of ψ.*

This result can in fact be used to show that the system converges to a clustering configuration. The clusters are determined by the orbits of the weak automorphism group. Here, one considers diffusively-coupled networks $(\mathcal{G}, \Sigma, \Pi)$ satisfying either Assumption 1 or 2. Under this assumption, the closed-loop network is known to converge, and the invariance properties of the steady-state limit are studied. Focusing on homogeneous networks, [16] further identified the value of $\gamma_e(0)$ as decisive. Namely, [16] shows that if $0 \in \gamma_e(0)$ for all $e \in \mathbb{E}$, then the closed-loop network $(\mathcal{G}, \Sigma, \Pi)$ converges to consensus, and otherwise it displays a clustering behavior. Namely, for homogeneous networks, two nodes $i, j \in \mathbb{V}$ are in the same cluster whenever $i, j \in \mathbb{V}$ are exchangable under the action of $\mathrm{Aut}(\mathcal{G})$, and the converse is almost surely true.

Although [16] presents an extensive study toward understanding the clustering behaviour of diffusively-coupled multi-agent systems, it does not offer a synthesis procedure toward solving the clustering problem:

Problem 1. *Consider a collection of n homogeneous agents $\{\Sigma_i\}_{i \in \mathbb{V}}$, and let $r_1, \ldots, r_k$ be positive integers which sum to n. Find a graph $\mathcal{G} = (\mathbb{V}, \mathbb{E})$ and homogeneous edge controllers $\{\Pi_e\}_{e \in \mathbb{E}}$ such that the closed loop network converges to a clustered steady-state, with a total of k clusters of sizes $r_1, \ldots, r_k$.*

The goal of this section is use the tools of [16] to solve Problem 1. As described above, this can be achieved in two steps. We first make the following assumption about the controllers:

Assumption 3. *The homogeneous MEIP controllers are chosen so that $0 \notin \gamma_e(0)$ for any edge $e \in \mathbb{E}$.*

Second, given the desired cluster sizes $r_1, \ldots, r_k$, we seek an oriented graph $\mathcal{G} = (\mathbb{V}, \mathbb{E})$ such that the action of $\mathrm{Aut}(\mathcal{G})$ on $\mathcal{G}$ has orbits of sizes $r_1, \ldots, r_k$. Moreover, we desire that the oriented graph $\mathcal{G}$ will be weakly connected[1] to assure a flow of information throughout the corresponding diffusively-coupled network. If we find such a graph, the results of [16] guarantee that the desired clustering behavior is achieved almost surely, so long that Assumption 3 holds. We make the following definition for the sake of brevity, and define the corresponding problem:

Definition 5. *The oriented graph $\mathcal{G}$ is said to be of type $OS(r_1, \ldots, r_k)$ if it is weakly connected and the action of $\mathrm{Aut}(\mathcal{G})$ on $\mathcal{G}$ has orbits of sizes $r_1, \ldots, r_k$.[2]*

Problem 2. *Given positive integers $r_1, \ldots, r_k$, determine if an oriented graph of type $OS(r_1, \ldots, r_k)$ exists, and if so, construct it.*

Additionally, running the system with an underlying graph $\mathcal{G}$ requires means to implement the corresponding interconnections. For that reason, graphs with fewer edges are more desirable. We wish to understand how many edges does a directed graph of type $OS(r_1, \ldots, r_k)$ require. This is formalized in the next result.

Theorem 2. *Let $r_1, \ldots r_k$ be any positive integers, and let $n = r_1 + \ldots + r_k$*

i) *Any directed graph of type $OS(r_1, \ldots, r_k)$ has at least m edges, where*

$$m = \min_{\mathcal{T} \text{ tree on } k \text{ vertices}} \sum_{\{i,j\} \in \mathcal{T}} \mathrm{lcm}(r_i, r_j). \quad (2)$$

ii) *There exist a directed graph of type $OS(r_1, \ldots, r_k)$ with at most M edges, where*

$$M = \min_{\substack{\mathcal{T} \text{ path on} \\ k \text{ vertices}}} \left(\sum_{\{i,j\} \in \mathcal{T}} \mathrm{lcm}(r_i, r_j) \right) + \min_{i \in \mathbb{V}} r_i. \quad (3)$$

Proof. We start with the former claim. Consider a graph $\mathcal{G}$ of type $OS(r_1, \ldots, r_k)$. Let $V_1, \cdots, V_k$ be the orbits of $\mathrm{Aut}(\mathcal{G})$ in $\mathcal{G}$, corresponding to the different clusters. For any two indices $i, j \in \{1, \cdots, k\}$, we consider the induced bi-partite subgraph $\mathcal{G}_{ij}$ on the vertices $V_i \cup V_j$.[3] We claim that if it is not empty, then $\mathcal{G}_{ij}$ has at least $\mathrm{lcm}(r_i, r_j)$ edges.

Indeed, V_j is invariant to $\mathrm{Aut}(\mathcal{G})$, hence, for any $x \in V_i$ and any $\psi \in \mathrm{Aut}(\mathcal{G})$, x and $\psi(x)$ have the same number

---

[1]Recall that a directed graph is weakly connected if its unoriented counterpart is connected.

[2]OS stands for "orbit structure"

[3]In other words, only edges between V_i and V_j exist in the subgraph.

of edges from them to V_j. Thus, all vertices in V_i have the same $\mathcal{G}_{ij}$-degree. Similarly, all vertices in V_j have the same $\mathcal{G}_{ij}$-degree. Let d_i be the $\mathcal{G}_{ij}$-degree of vertices in V_i, and d_j be the $\mathcal{G}_{ij}$-degree of vertices in V_j. As the edges in $\mathcal{G}_{ij}$ only link V_i and V_j, the number of edges in $\mathcal{G}_{ij}$ is equal to $r_i d_i = r_j d_j$. In particular, the number r_j divides $r_i d_i$, and the number $\frac{r_j}{\gcd(r_i, r_j)}$ divides $\frac{r_i}{\gcd(r_i, r_j)} d_i$. However, the numbers $\frac{r_j}{\gcd(r_i, r_j)}, \frac{r_i}{\gcd(r_i, r_j)}$ are relatively prime, meaning that $\frac{r_j}{\gcd(r_i, r_j)}$ must divide d_i. In particular, $d_i \geq \frac{r_j}{\gcd(r_i, r_j)}$, and $\mathcal{G}_{ij}$ has at least $r_i d_i \geq \frac{r_i r_j}{\gcd(r_i, r_j)} = \mathrm{lcm}(r_i, r_j)$ edges.

Now, consider the condensed graph $\mathcal{G}'$. The vertices of $\mathcal{G}'$ are $\{v_1, \cdots, v_k\}$, and v_i is connected to v_j if and only if there is an edge between V_i and V_j. As $\mathcal{G}$ is weakly connected, $\mathcal{G}'$ is also connected. Let $\mathcal{T}$ be a spanning tree for $\mathcal{G}'$. If there is an edge $e = (i, j)$ in $\mathcal{T}$, V_i and V_j are linked, meaning that the graph $\mathcal{G}_{ij}$ contains at least $\mathrm{lcm}(r_i, r_j)$ edges. As a result, $\mathcal{G}$ has at least $\sum_{\{i,j\} \in \mathcal{T}} \mathrm{lcm}(r_i, r_j) \geq m$ edges.

We now consider the second part of the theorem. Let $\mathcal{T}$ be a path on k vertices, and let $i_1, \cdots, i_k$ be the order of the vertices in the path. By reordering $r_1, \cdots, r_k$, we may assume that $i_j = j$ for $j = 1, 2, \cdots, k$. Let i be the vertex at which r_i is minimized. For each $j \in \{1, \cdots, k\}$, we number the vertices in V_j as $v_1^j, \cdots, v_{r_j}^j$. Build the graph $\mathcal{G}$ as follows:

I) For every j, add the following edges between V_j and V_{j+1}: For $p = 1, \ldots, \mathrm{lcm}(r_j, r_{j+1})$, connect $v_{p \mod r_j}^j$ to $v_{p \mod r_{j+1}}^{j+1}$.

II) If $r_i > 1$, then for each $p = 1, \ldots, r_i$, connect v_p^i to $v_{(p+1) \mod r_i}^i$.

The number of edges in $\mathcal{G}$ is equal to $\sum_{\{i,j\} \in \mathcal{T}} \mathrm{lcm}(r_i, r_j) + r_i$ if $r_i \geq 2$, and equal to $\sum_{\{i,j\} \in \mathcal{T}} \mathrm{lcm}(r_i, r_j)$ otherwise. We show that the orbits of the action of $\mathrm{Aut}(\mathcal{G})$ on $\mathcal{G}$ are given by $V_1, \cdots, V_k$, and that G is weakly connected, concluding the proof.

First, we show $\mathcal{G}$ is weakly connected. We note that the induced subgraph on V_i is weakly connected. Indeed, this is clear if $r_i = 1$, and in the case $r_i \geq 2$, the following cycle eventually passes through all the nodes in V_i:

$$v_1^i \to v_2^i \to v_3^i \to \cdots$$

Now, by the construction of $\mathcal{G}$, any vertex v_p^i is connected all vertices $v_{p \mod r_l}^l$. Thus, for any two arbitrary vertices $v_{p_1}^{j_1}$ and $v_{p_2}^{j_2}$, we can consider a path that starts from $v_{p_1}^{j_1}$, goes to $v_{p_1 \mod r_i}^i$, moves to $v_{p_2 \mod r_i}^i$ using the connectivity of induced subgraph on V_i, and continues to $v_{p_2}^{j_2}$. Thus $\mathcal{G}$ is weakly connected.

As for the orbits, we first consider the map ψ defined on the vertices of $\mathcal{G}$ by sending each vertex v_p^j to $v_{(p+1) \mod r_j}^j$. By definition of the graph $\mathcal{G}$, it is clear that ψ is a graph automorphism. Moreover, repeatedly applying ψ can move v_p^j to any vertex in V_j. Thus the orbit of v_p^j contains the set V_j.

Conversely, we show that each V_j is invariant under $\mathrm{Aut}(\mathcal{G})$, so the orbits of the action of $\mathrm{Aut}(\mathcal{G})$ on $\mathcal{G}$ are exactly $V_1, \cdots V_k$. This is obvious if $k = 1$, as then $V_1 = \mathbb{V}$, so we assume that $k \geq 2$. In that case, either $i \neq 1$ or $i \neq k$

(or both). We assume $i \neq k$, as the complementary case can be treated similarly. Graph automorphisms preserve all graph properties, and in particular, they preserve the out-degree of vertices. As all edges are oriented from V_j to V_{j+1} or from V_i to itself where $i \neq k$, the vertices in V_k have an out-degree of 0, and they are the only ones with this property. Thus V_k must be invariant under $\mathrm{Aut}(\mathcal{G})$. The only vertices with edges to V_k are in V_{k-1}, meaning that V_{k-1} is also invariant under $\mathrm{Aut}(\mathcal{G})$. Iterating this argument, we conclude that $V_1, \cdots V_k$ must all be invariant under the action of $\mathrm{Aut}(\mathcal{G})$. Thus they are the orbits of the action, and the action of $\mathrm{Aut}(\mathcal{G})$ on $\mathcal{G}$ has orbits of size $r_1, \cdots, r_k$. This completes the proof. $\square$

The construction of OS-type graphs presented in the proof is explicitly stated in Algorithm 1. The Algorithm takes as input the desired cluster sizes, as well as a path $\mathcal{T}$ describing the desired interconnection topology of the different clusters in the graph. While any path gives a graph of type $OS(r_1, \ldots, r_k)$, the number of edges in the graph depends on the path $\mathcal{T}$.

Algorithm 1 Building OS-type graphs

Input: A collection $r_1, \ldots, r_k$ of positive integers summing to n, and a path $\mathcal{T}$ on k vertices.
Output: A graph $\mathcal{G}$ of type $OS(r_1, \ldots, r_k)$.
1: Let $\mathcal{G} = (\mathbb{V}, \mathcal{E})$ be an empty graph.
2: **for** $j = 1, \ldots, k$ and $p = 1, \ldots, r_j$ **do**
3:      Add a node with label v_p^j to $\mathbb{V}$.
4: **for** any edge $\{i, j\}$ in $\mathcal{T}$ **do**
5:      **for** $p = 1, \ldots, \mathrm{lcm}(r_i, r_j)$ **do**
6:          Add the edge $v_{p \mod r_i}^i \to v_{p \mod r_j}^j$ to $\mathbb{E}$.
7: Compute $i^\star = \arg \min\{r_i\}$. If $r_{i^\star} = 1$, go to step 10.
8: **for** $p = 1, \ldots, r_{i^\star}$ **do**
9:      Add the edge $v_p^{i^\star} \to v_{(p+1) \mod r_{i^\star}}^{i^\star}$ to $\mathbb{E}$.
10: **Return** $\mathcal{G} = (\mathbb{V}, \mathbb{E})$.

Remark 1. *One might ask why the lower bound considers all possible trees, while the upper bound only considers path graphs. The main reason for this distinction can be seen in the proof - one can build a general graph $\mathcal{G}$ using a tree graph as a basis, instead of the path graph $1 \to 2 \to 3 \to \cdots \to k$. In that case, proving that the orbit of v_p^j contains V_j is easy, but it is possible that the orbit is actually larger than V_j.*

Remark 2. *The lower bound in Theorem 2 can be found using any algorithm finding a minimal spanning tree, e.g. Kruskal's algorithm which runs in polynomial time [24]. However, the upper bound requires one to solve a variant of the traveling salesman problem, which is known to be **NP**-hard [24].*

Theorem 2 deals with a general cluster assignment by explicitly constructing a graph solving the problem. We apply it to more specific cases in order to achieve concrete bounds on the number of edges needed for clustering in these cases.

Corollary 1. *Suppose that all cluster sizes $r_1, \cdots, r_k$ are equal, and bigger than 1. Then there exists a graph of type $OS(r_1, \ldots, r_k)$ with at most $n = r_1 + \ldots + r_k$ edges.*

Proof. Let r be the size of all clusters. We note that in this case, $\text{lcm}(r, r) = r$, and the number of clusters is $k = n/r$. Thus, if we build a graph $\mathcal{G}$ as in the proof of Theorem 2, then $\mathcal{G}$ has exactly n edges, as the summation over the edges has $k - 1$ elements. It remains to show that no such graph with less than n edges exists.

First, any graph on n nodes and less than $n - 1$ edges is not weakly connected [25]. Thus, it suffices to show that no such graph with $n - 1$ edges exists. Indeed, let d_i be the out-degree of any vertex in the i-th orbit, $i = 1, \cdots, k$. As the action of $\text{Aut}(\mathcal{G})$ preserves the out-degree, it is the same for all vertices in the i-th orbit. The number of edges is the sum of the out-degree over all vertices, but it's equal to $rd_1 + \cdots + rd_k$. Thus the number of edges in $\mathcal{G}$ must be divisible by r. But $n = kr$, meaning that $n - 1$ is not divisible by r unless $r = 1$. Thus there is no such graph on $n - 1$ edges. $\square$

Corollary 2. *Let $r_1, \cdots, r_k$ be positive integers such that $k \geq 2$ and that for every j, l, either r_j divides r_l or vice versa. Then there exists a graph of type $OS(r_1, \ldots, r_k)$ with $n = r_1 + \ldots + r_k$ edges.*

Proof. We reorder the numbers $r_1, \cdots, r_k$ such that r_l divides r_j for $l \leq j$. We note that if r_l divides r_j, then $\text{lcm}(r_l, r_j) = r_l$. Thus, if we let $\mathcal{G}$ be the graph built in the proof of Theorem 2, it has the following number of edges:

$$\sum_{j=1}^{k-1} \text{lcm}(r_j, r_{j+1}) + r_1 = \sum_{j=1}^{k-1} r_{j+1} + r_1 = \sum_{j=1} r_j = n.$$

$\square$

Corollary 3. *Let $r_1, \cdots, r_k$ be positive integers such that $r_j \leq q$ for all j, and let $n = r_1 + \cdots + r_k$. Then there exists a graph of type $OS(r_1, \ldots, r_k)$ with at most $n + O(q^3)$ edges.*

Proof. By reordering, we may assume that that $r_1 \leq r_2 \leq \cdots r_k$. Let m_l be the number of clusters of size s for $l = 1, 2, \cdots, q$. As before, consider the graph $\mathcal{G}$ built in the proof of Theorem 2. If $r_j = r_{j+1}$ then $\text{lcm}(r_j, r_{j+1}) = r_l$, and $\text{lcm}(r_j, r_{j+1}) \leq r_j r_{j+1}$ otherwise. Thus, the number of edges in $\mathcal{G}$ is given by:

$$\sum_{j=1}^{k-1} \text{lcm}(r_j, r_{j+1}) + r_1 \leq \sum_{\substack{l \in \{1, \ldots, q\}, \\ m_l \neq 0}} (m_l - 1)l + \sum_{l=1}^{q-1} l(l-1) + r_1.$$

Indeed, for each $l \in \{1, \ldots, q\}$, if there's at least one cluster of size l, then there are $m_l - 1$ edges in the path $\mathcal{T} = 1 \rightarrow 2 \rightarrow \cdots \rightarrow k$ that touch two clusters of size l. The second term bounds the number of edges that appear between clusters of different sizes. We note that $n = \sum_{l=1}^{q} lm_l$, so the first term is bounded by n. As for the second term, we can bound $l(l-1)$ by l^2 and then use the formula $\sum_{l=1}^{q-1} l^2 = \frac{(q-1)q(2q-1)}{6}$, dating back to Fibonacci's *Liber Abaci* from 1201 [26]. Lastly, the last term r_1 is bounded by q. This completes the proof. $\square$

We now give examples of graphs constructed by Algorithm 1, and show they indeed force a clustering structure on the agents.

Example 1. *We consider a collection of identical agents, all of the form $\dot{x} = -x + u + \alpha, y = x$ where α is a log-uniform random variable between 0.1 and 1, identical for all agents. In all experiments described below, we considered identical controllers, equal to the static nonlinearity of the form*

$$\mu = a_1 + a_2(\zeta + \cos(\zeta)),$$

where a_1 was chosen as a Gaussian random variable with mean 0 and standard deviation 10, and a_2 was chosen as a log-uniform random variable between 0.1 and 10. We note that the agents are indeed output-strictly MEIP and the controllers are MEIP. Moreover, the network is homogeneous, so $\text{Aut}(\mathcal{G}, \Sigma, \Pi) = \text{Aut}(\mathcal{G})$. Thus, we can use the graphs constructed by Theorem 2, as depicted in Algorithm 1, to force a clustering behavior.

We first consider a network of $n = 15$ agents and tackle the cluster synthesis problem with five equally-sized clusters, i.e., $r_1 = r_2 = r_3 = r_4 = r_5 = 3$. One possible graph forcing these clusters, as constructed by Algorithm 1 for a path $\mathcal{T}$ of the form $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$, can be seen in Figure 2, along with the agents' trajectories for the closed-loop system.

Second, we consider a network of $n = 10$ agents with desired cluster sizes $r_1 = 1, r_2 = 2, r_3 = 3, r_4 = 4$. We build a graph forcing these cluster sizes by using Algorithm 1 for a path $\mathcal{T}$ of the form $4 \rightarrow 2 \rightarrow 1 \rightarrow 3$, which is the minimizer in (3). The graph can be seen in Figure 2, along with the agents' trajectories for the closed-loop system.

IV. Conclusions

In this work we explored the problem of cluster assignment for homogeneous multi-agent systems. We extended the results of [16] to provide a characterization of graphs with an automorphism group containing orbits of a prescribed size. When such graphs are used in a network comprised of weakly equivalent agent and controller dynamics, the network converges to a cluster configuration.

References

[1] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65 – 78, 2004.

[2] N. Chopra and M. W. Spong, *Advances in Robot Control: From Everyday Physics to Human-Like Movements*. Springer, 2006, ch. Passivity-Based Control of Multi-Agent Systems, pp. 107–134.

[3] L. Scardovi, M. Arcak, and E. Sontag, "Synchronization of interconnected systems with applications to biochemical networks: An input-output approach," *IEEE Transactions on Automatic Control*, vol. 55, no. 6, pp. 1367–1379, Jun. 2010.

[4] R. Olfati-Saber, "Distributed kalman filtering for sensor networks," in *2007 46th IEEE Conference on Decision and Control*, 2007, pp. 5492–5498.

[5] A. Schnitzler and J. Gross, "Normal and pathological oscillatory communication in the brain," *Nature reviews. Neuroscience*, vol. 6, pp. 285–96, 05 2005.

[6] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 52–67, June 2002.
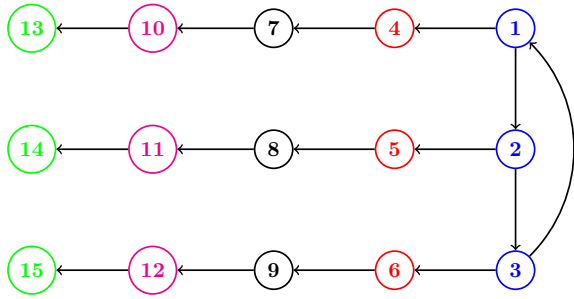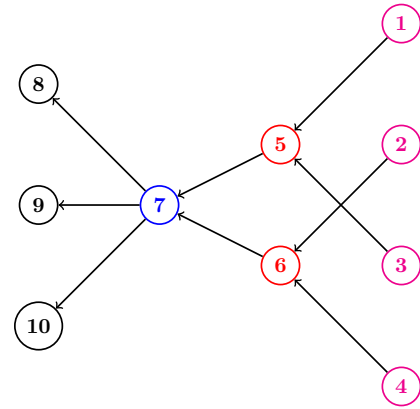
(a) Graph forcing cluster sizes $r_1 = r_2 = r_3 = r_4 = r_5 = 3$. Nodes with the same color will be in the same cluster.
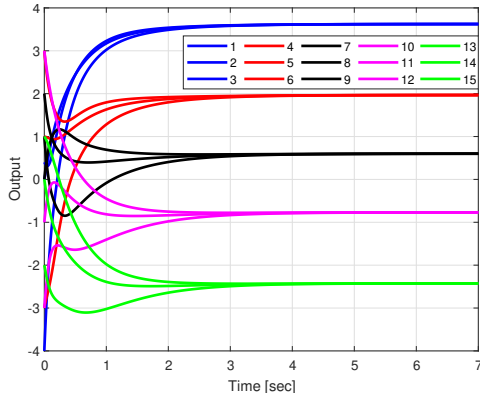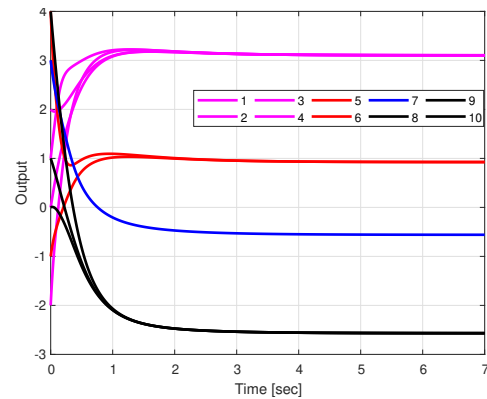

(b) Agent's trajectories for the closed-loop system. Colors correspond to node colors in the graph.

Fig. 2. First example of graphs solving the cluster synthesis problem, achieved by running Algorithm 1.

(a) Graph forcing cluster sizes $r_1 = 1, r_2 = 2, r_3 = 3, r_4 = 4$. Nodes with the same color will be in the same cluster.

(b) Agent's trajectories for the closed-loop system. Colors correspond to node colors in the graph.

Fig. 3. First example of graphs solving the cluster synthesis problem, achieved by running Algorithm 1.

[7] A. Lancichinetti and S. Fortunato, "Consensus clustering in complex networks," *Scientific Reports*, vol. 2, no. 1, p. 336, dec 2012.

[8] C. Altafini, "Consensus problems on networks with antagonistic interactions," *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 935–946, 2013.

[9] J. Qin and C. Yu, "Cluster consensus control of generic linear multi-agent systems under directed topology with acyclic partition," *Automatica*, vol. 49, no. 9, pp. 2898 – 2905, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109813003312

[10] Y. Han, W. Lu, and T. Chen, "Cluster consensus in discrete-time networks of multiagents with inter-cluster nonidentical inputs," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 4, pp. 566–578, April 2013.

[11] M. Bürger, D. Zelazo, and F. Allgöwer, "Hierarchical Clustering of Dynamical Networks Using a Saddle-Point Analysis," *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 113–124, jan 2013.

[12] A. Rahmani and M. Mesbahi, "Pulling the strings on agreement: Anchoring, controllability, and graph automorphisms," in *Proc. 2007 American Control Conference*, July 2007, pp. 2738–2743.

[13] A. Chapman and M. Mesbahi, "On symmetry and controllability of multi-agent systems," in *Proc. 53rd IEEE Conference on Decision and Control*, Dec 2014, pp. 625–630.

[14] ——, "State controllability, output controllability and stabilizability of networks: A symmetry perspective," in *Proc. 2015 54th IEEE Conference on Decision and Control (CDC)*, Dec 2015, pp. 4776–4781.

[15] G. Notarstefano and G. Parlangeli, "Controllability and observability of grid graphs via reduction and symmetries," *IEEE Transactions on Automatic Control*, vol. 58, no. 7, pp. 1719–1731, 2013.

[16] M. Sharf and D. Zelazo, "Symmetry-induced clustering in multi-agent systems using network optimization and passivity," in *Proc. 2019 27th Mediterranean Conference on Control and Automation (MED)*, July 2019, pp. 19–24.

[17] C. Godsil and G. Royle, *Algebraic Graph Theory*, ser. Graduate Texts in Mathematics. Springer New York, 2001.

[18] M. Bürger, D. Zelazo, and F. Allgöwer, "Duality and network theory in passivity-based cooperative control," *Automatica*, vol. 50, no. 8, pp. 2051–2061, 2014.

[19] G. H. Hines, M. Arcak, and A. K. Packard, "Equilibrium-independent passivity: A new definition and numerical certification," *Automatica*, vol. 47, no. 9, p. 1949–1956, 2011.

[20] M. Sharf and D. Zelazo, "A network optimization approach to cooperative control synthesis," *IEEE Control Systems Letters*, vol. 1, no. 1, pp. 86–91, July 2017.

[21] A. Jain, M. Sharf, and D. Zelazo, "Regulatization and feedback passivation in cooperative control of passivity-short systems: A network optimization perspective," *IEEE Control Systems Letters*, vol. 2, pp. 731–736, 2018.

[22] M. Sharf, A. Jain, and D. Zelazo, "A geometric method for passivation and cooperative control of equilibrium-independent passivity-short systems," *IEEE Transactions on Automatic Control*, vol. 66, no. 12, pp. 5877–5892, 2021.

[23] D. S. Dummit and R. M. Foote, *Abstract algebra*. Prentice Hall Englewood Cliffs, NJ, 1991, vol. 1999.

[24] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Third Edition*, 3rd ed. The MIT Press, 2009.

[25] J. A. Bondy, U. S. R. Murty *et al.*, *Graph theory with applications*. Macmillan London, 1976, vol. 290.

[26] L. Sigler, *Fibonacci's Liber Abaci: a translation into modern English of Leonardo Pisano's book of calculation*. Springer Science & Business Media, 2003.