Distributed Design of Strong Structurally Controllable and Maximally Robust Networks

Priyanshkumar I. Patel*, Johir Suresh*, and Waseem Abbas

Abstract—This paper studies the problem of designing multiagent networks that simultaneously achieve strong structural controllability (SSC) and maximal robustness. Though crucial for effective operation, these two properties can conflict in multiagent systems. We present novel methods to construct network graphs that balance these competing requirements while accounting for network parameters such as the total number of agents N and the number of leaders N_L (agents utilized to inject external inputs into the network). We further extend our framework to incorporate the network diameter D, thereby generating both maximally robust and strong structurally controllable networks for given parameters N, N_L, and D. To assess controllability, we employ the notion of zero forcing sets in graphs and rigorously evaluate the robustness of our designs. We also present a distributed approach to constructing these networks, leveraging graph grammars. This work explores the trade-off between network controllability and robustness to achieve multiple design objectives in multiagent systems.

Keywords—Strong structural controllability, zero forcing sets, network design, network robustness.

1 Introduction

A recurring challenge in network design resides in accommodating a range of distinct and occasionally conflicting design criteria. Designing distributed networks possessing desired attributes while simultaneously adhering to a multitude of constraints underscores a substantial challenge. From a network control perspective, *controllability* and *robustness* are crucial design attributes. Network controllability is the ability to drive the network to desired configurations (states), achieved via external control signals (inputs) injected into the network through a subset of nodes called *leaders*. On the other hand, network robustness assumes distinct forms as functional and structural robustness [2]. The former relates to the network's performance in the face of noise and perturbations, while the latter denotes its ability to preserve structural attributes despite node or edge failures [3]–[5]. Remarkably, these interpretations are interlinked within the context of network control systems and can be quantified harmoniously through standard graph metrics, such as the Kirchhoff index K_f [2], [6], [7].

The intricate interplay between network controllability and robustness has been well studied, suggesting that these attributes can often be opposing to each other: networks achieving complete controllability with minimal leaders for given network parameters might exhibit subpar robustness [8], [9]. For instance, if the number of agents N is a network parameter, the network achieving complete controllability through a single leader agent is a path network; however, path networks have minimum robustness. Similarly, when the number of agents N and the diameter D are set, networks with the maximum robustness (as measured by the Kirchhoff index K_f) are clique chains [3], [6]; however, they demonstrate poor controllability in the sense that many (precisely, $N - D$) leaders are needed to make such networks completely controllable [10]. This presents an intriguing question: *how can we design networks in a distributed manner such that networks are controllable with few leaders (inputs) and concurrently exhibit high robustness?* This question becomes more challenging when the network controllability is considered in the strong structural sense due to computational complexity intricacies (e.g., [11]–[13]). Network controllability generally depends on the edge weights; however, edge weights are inconsequential for strong structural controllability (SSC), which essentially depends on the network structure and the leader set (as Section 2.2 explains). This distinction holds significant value in real-world scenarios where obtaining precise edge weight information can be challenging due to physical constraints, noise, or imprecise system parameters.

This paper presents an innovative approach to constructing graphs that achieve dual objectives: strong structural controllability and maximal robustness, all within the confines of predefined graph parameters. Initially, we present constructions tailored to the two parameters, namely the number of nodes N and the number of leaders N_L, while generating graphs that are provably strong structurally controllable and robust due to the maximal edge sets. We further extend these constructions to encompass another design parameter: the graph diameter D, often a design requirement in practical networks. To ensure strong structural controllability, we leverage the compelling relationship between the concept of zero forcing in graphs and SSC [14], [15]. Zero forcing is a node coloring process in graphs initiated by a subset of initially colored nodes that subsequently color other nodes according to some predefined rules. Crucially, a zero forcing set encompasses the subset of nodes that, when initially colored, inevitably leads to the coloring of the entire graph. Monshizadeh et al. established in [14] that the network whose leader nodes constitute a zero

*Authors contributed equally to the work.
Authors are with the University of Texas at Dallas, Richardson, TX, USA (Emails: priyanshkumar.patel@utdallas.edu, johir.suresh@utdallas.edu, waseem.abbas@utdallas.edu).
Some preliminary results appeared in [1].

forcing set is strong structurally controllable. Building upon this insight, our constructions carefully select N_L leaders in a graph to form a zero forcing set of the graph, thereby ensuring SSC. Finally, we discuss a distributed construction of such graphs using the idea of graph grammars [16]–[18]. Our main contributions are summarized below:

- For given N (total number of nodes) and N_L (number of leaders), we construct strong structurally controllable graphs with N_L leaders and maximal edge sets, i.e., no additional edges can be added without compromising SSC. The graphs are carefully designed to ensure that the leader set constitutes a zero forcing set, thereby ensuring SSC. This approach results in graphs with fixed diameters: either $D = N/N_L$ or 2. Section 3 presents the details of these designs.
- We then extend our method to accommodate an additional input: the graph diameter D. With N, N_L, and D as parameters, we construct graphs that achieve both SSC and maximal robustness. Our approach ensures that leader sets form zero forcing sets, and the resulting graphs are systematically analyzed for their robustness using the Kirchhoff index measure. We also compare the designed graphs against clique chains, a graph class known for optimal robustness under given N and D parameters. Section 4 elaborates on this design approach and the corresponding analysis.
- For practicality and scalability, We also propose a distributed method to construct such graphs leveraging the idea of graph grammars, enabling nodes to locally implement a set of rules that collectively shape the desired network structure (Section 5). We also numerically evaluate the proposed design schemes.

Our problem setting is similar to the one in [8], albeit with some significant differences. Instead of relying on graph distances as in [8], we employ a simpler zero-forcing method to examine strong structural controllability within networks. The rationale behind this choice is that the distance-based bound always requires an equal or more number of leaders than the zero-forcing method to guarantee SSC [19]. Additionally, the computational complexity of the zero-forcing bound is better than the distance-based bound. In particular, for a given set of N_L leader nodes in a graph with N nodes, the zero forcing bound can be computed in $O(N + N_L)$ time [20]. On the other hand, the distance-based bound needs $O(N_L(N \log N + N^{N_L}))$ time for the exact computation and $O(N_L N \log N)$ for the approximate computation [12]. Further, for given N and N_L, the graphs generated in [8] are of fixed diameter, however, we introduce multiple construction methods enabling the generation of graphs with varying diameters and degrees of robustness. Finally, we also provide distributed constructions of networks using graph grammars, which are not in [8]. Some other works also address the improvement of the given network's robustness through edge augmentation while accounting for the network controllability [21]–[27]. Unlike most of these works, our paper focuses on the *design problem*, which entails generating graphs with predefined parameters and concurrently achieving both SSC and maximal robustness. We also note that, our paper primarily addresses undirected graphs. Nevertheless, the approach can be extended to directed graphs with appropriate modifications to the zero forcing method and the notion of Kirchhoff index [5], [28], [29], allowing for adaptation to the directed case. The resulting directed graphs, however would differ from the designed networks presented here.

The rest of the paper is organized as follows: Section 2 presents preliminary ideas, discusses SSC in networks using zero forcing, and elaborates on the network robustness. Section 3 presents strong structurally controllable and robust graph designs for fixed N and N_L. Section 4 also includes network diameter D as an input in the design process and details general graph designs with robustness analysis. Section 5 provides graph grammars to construct the proposed graphs. Finally, Section 6 concludes the paper.

2 PRELIMINARIES

2.1 Notations

We consider a multi-agent network modeled by an *undirected graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The *vertex set* $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ represents the agents, also referred to as nodes, and the edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents the interactions between nodes. We denote the edge between nodes u and v by an unordered pair (u, v). Node u is a *neighbor* of node v if $(u, v) \in \mathcal{E}$. The number of nodes in the neighborhood of u is the *degree* of u. The *adjacency matrix* of graph $\mathcal{G}$ with n nodes is denoted by $\mathcal{A} \in \mathbb{R}^{n \times n}$. It is a matrix whose ij^{th} entry $(\mathcal{A}_{ij})$ is 1 if nodes v_i and v_j are neighbors, and 0 otherwise. The *degree matrix* of a graph, denoted by Δ, is a diagonal matrix in which each diagonal entry Δ_{ii} corresponds to the degree of node v_i. Similarly, the *Laplacian matrix* of $\mathcal{G}$, denoed by $\mathcal{L}$ is simply $\Delta - \mathcal{A}$. The *distance* between nodes u and v, denoted by $d(u, v)$, is the number of edges in the shortest path between u and v. The *diameter* of $\mathcal{G}$, denoted by D, is the maximum distance between any two nodes in $\mathcal{G}$. A *path* of length k in $\mathcal{G}$ is a sequence of nodes, $P_k :=< u_0, u_1, u_2, \cdots, u_k >$, where $(u_i, u_{i+1}) \in \mathcal{E},\ \forall i \in \{0, \cdots, k-1\}$. The *leader - follower* system associated with graph $\mathcal{G}$ is defined by the following:

$$\dot{x}(t) = Mx(t) + Bu(t). \tag{1}$$

Here, $x(t) = \begin{bmatrix} x_1(t) & \cdots & x_n(t) \end{bmatrix}^\intercal \in \mathbb{R}^n$ is the state of the system at time t, and $x_i(t) \in \mathbb{R}$ is the state of node v_i at time t. The matrix $M \in \mathbb{R}^{n \times n}$ is the system matrix and belongs to a family of symmetric matrices $\mathcal{M}(\mathcal{G})$ associated with $\mathcal{G}$. We define $\mathcal{M}(\mathcal{G})$ as,

$$\begin{aligned} \mathcal{M}(\mathcal{G}) = \{M \in \mathbb{R}^{n \times n} | M = M^T, \text{and for } i \neq j, \\ M_{ij} \neq 0 \Leftrightarrow (i, j) \in \mathcal{E}(\mathcal{G})\}. \end{aligned} \tag{2}$$

Note that $\mathcal{M}(\mathcal{G})$ includes the adjacency and Laplacian matrices of $\mathcal{G}$. In (1), $u(t) \in \mathbb{R}^m$ is the input signal, and $B \in \mathbb{R}^{n \times m}$ is the input matrix containing information about the *leader* nodes, which are the nodes through which inputs are injected into the network. For a set of m leaders labelled $\{\ell_1, \ell_2, \ldots, \ell_m\} \subseteq \mathcal{V}$, we define the input matrix as follows.

$$B_{ij} = \begin{cases} 1 & \text{if } v_i = \ell_j, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

We are interested in designing networks following (1) that are simultaneously controllable and maximally robust. Next, we discuss the controllability and robustness measures used throughout the paper.

2.2 Network Controllability

We utilize the concept of strong structural controllability (SSC) for network controllability, which determines whether a given network can be controlled to reach any desired state from any initial state within a given time period. More precisely, consider a system (1) defined on graph $\mathcal{G}$, the pair (M, B) is a *controllable pair* if there exists an input $u(t)$ that can drive the system from an arbitrary initial state $x(t_0)$ to any final state $x(t_f)$ in a given time $t = t_f - t_0$.

Definition 2.1. (Strong Structural Controllability (SSC)) *A given graph $\mathcal{G}$ with a set of leader nodes $\{\ell_1, \ell_2, \ldots, \ell_m\} \subseteq \mathcal{V}$, and the corresponding B matrix is said to be* strong structurally controllable *if and only if (M, B) is a controllable pair for all $M \in \mathcal{M}$.*

Note that M_{ij} (i.e., the ij^{th} entry of M) in (1) represent the weight of the edge between nodes v_i and v_j, signifying the coupling between these two nodes. Remarkably, the SSC of $\mathcal{G}$ implies that the network is controllable for one set of (non-zero) edge weights if and only if it remains controllable for any other set of (non-zero) edge weights (as in (2)). As a result, the SSC of $\mathcal{G}$ depends on the structure and not the edge weights. Since capturing precise values of edge weights is generally very difficult due to uncertainties, numerical inaccuracies, and inexact measurement of system parameters, the notion of SSC is particularly valuable in such scenarios. Strong structural controllability ensures that the network remains controllable even in worse situations where edge weights are unknown or might perturb. The next step is to determine how we can ascertain whether a given network is strong structurally controllable, or not.

In [14], Monshizadeh et al. provide an elegant graph-theoretic characterization of strong structural controllability in networks utilizing a graph coloring process called the *zero forcing* in graphs, which we describe below.

Definition 2.2. (Zero Forcing Process (ZFP)) *Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where the nodes are initially colored either black or white, and the ZFP is applied as follows: If a black node $v \in \mathcal{V}$ has exactly one white neighbor u, then v forces u to change its color to black. This color change rule is iteratively applied until no black node exists with only one white neighbor, completing the ZFP.*

For a given set of initial black nodes, there can be multiple ways to execute the ZFP; however, the set of black nodes at the end of the process will always be the same [30]. If there is a *unique* way of proceeding with the ZFP in a graph $\mathcal{G}$, we call it a *unique zero forcing process*. Moreover, the set of black nodes obtained at the conclusion of the ZFP is referred to as the *derived set*. We define a relevant and useful notion of zero forcing set below.

Definition 2.3. (Zero Forcing Set (ZFS)) *Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with an initial set of black nodes (leaders) $\{\ell_1, \ell_2, \ldots, \ell_m\} \subseteq \mathcal{V}$. Let $\mathcal{V}'$ be the derived set at the end of the zero forcing process, then $\{\ell_1, \ell_2, \ldots, \ell_m\}$ is a ZFS if and only if $\mathcal{V}' = \mathcal{V}$.*

Figure 1 illustrates the idea of a ZFS.

Monshizadeh et al. [14] characterizes the *network SSC in terms of the zero forcing process*. They show the following:

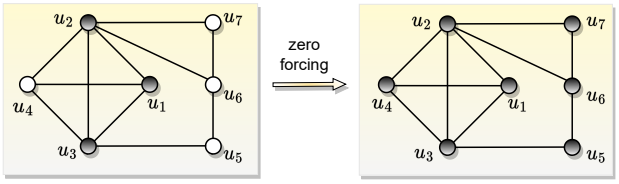


Fig. 1: Set of nodes $\{u_1, u_2, u_3\}$ is a ZFS as the corresponding derived set $\mathcal{V}'$ contains all the nodes in the graph.

Theorem 2.1. *[14], [31] An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a leader set $\mathcal{V}_L \subseteq \mathcal{V}$ is strong structurally controllable (as defined in Defintion 2.1) if and only if $\mathcal{V}_L$ is a ZFS of $\mathcal{G}$.*

A direct corollary of the above result is that the minimum number of leaders required to make $\mathcal{G}$ strong structurally controllable is equal to the *zero forcing number* of $\mathcal{G}$, which is the size of the smallest ZFS of $\mathcal{G}$. This work aims to design strong structurally controllable networks with a given number of leader nodes $N_L \leq |\mathcal{V}|$. Hence, *our design entails selecting a set of leaders constituting a ZFS of $\mathcal{G}$.*

2.3 Network Robustness

To analyze the robustness of the proposed graphs, we use a widely used metric, *Kirchhoff index*, which is denoted by $K_f(\mathcal{G})$ and defined as,

$$K_f(\mathcal{G}) = n \sum_{i=2}^{n} \frac{1}{\lambda_i}, \tag{4}$$

where n is the total number of nodes in the graph $\mathcal{G}$ and $\lambda_2 \leq \lambda_3 \leq \cdots \lambda_n$ are the (non-zero) eigenvalues of the Laplacian of $\mathcal{G}$. The Kirchhoff index has proven to be an effective measure for quantifying various aspects of robustness in networked dynamical systems. Notably, it has been extensively utilized to assess functional robustness, which measures the network's ability to mitigate the impact of noise on its performance. This includes considerations such as the expected population variance in steady-state due to noise (e.g., [5], [28], [32]–[35]). Additionally, Kirchhoff index measures structural robustness by evaluating the effect of node or link removal on the overall graph structure (e.g., [2], [3], [8], [32], [36], [37]). Importantly, network robustness, as measured by the Kirchhoff index, is a monotonically increasing function of edge additions. This implies that augmenting edges in a graph enhances its robustness to failures and noise. A smaller $K_f(\mathcal{G})$ indicates improved robustness to structural perturbations and noise, and vice versa. Therefore, our objective for enhancing robustness is to *maximize the number of edges in a graph* while considering other design constraints.

3 Controllable and Robust Networks

This section explores the relationship between network controllability and robustness while considering the measures discussed in Sections 2.2 and 2.3. We aim to design networks that are simultaneously strong structurally controllable and maximally robust given various network design parameters, including the total number of nodes N and the number of leader nodes N_L. Subsequently, we will conduct extensive

experimentation to analyze and compare the performance of our proposed network designs. Finally, in Section 5, we will present a distributed approach to construct the proposed graphs, wherein each node follows a set of local rules, i.e., *graph grammars*, to make connections with other nodes and achieve the desired structure. The main design problem discussed in this section is stated below.

Problem 1: Given the total number of nodes N and the number of leaders N_L, how can we design graphs that are strong structurally controllable and maximally robust?

Our goal is to select the maximal edge set, representing interconnections and information exchange among agents, in the underlying network graph of N agents, and select a set of N_L leader nodes rendering the graph strong structurally controllable. We present two solutions that differ in terms of the resulting graph's diameter D and then, in Section 4, we also introduce a method to combine these designs to create graphs with varying diameters.

3.1 Network Design 1

In this subsection, we construct a graph $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ with a leader set $\mathcal{V}_L \subseteq \mathcal{V}_1$. We consider $|\mathcal{V}_1| = N$ and $|\mathcal{V}_L| = N_L$, where N and N_L are the given number of nodes and leaders, respectively. Our construction ensures that $\mathcal{G}_1$ has a maximal edge set, and leaders constitute a ZFS of $\mathcal{G}_1$, guaranteeing that $\mathcal{G}_1$ is strong structurally controllable. We also show that the resulting graph $\mathcal{G}_1$ will have a diameter $D = \left\lceil \frac{N}{N_L} \right\rceil$.

For $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$, consider the following vertex set:

$$\mathcal{V}_1 = \{\ell_i\} \cup \{u_{i,j}\},$$

where $i \in \{1, 2, \ldots, N_L\}$ and $j \in \{1, 2, \ldots, D-1\}$. Nodes with labels $\{\ell_1, \ell_2, \ldots, \ell_{N_L}\}$ are leaders, whereas nodes with labels $\{u_{1,1}, \cdots, u_{N_L,D-1}\}$ are followers. We adjoin nodes in the following manner and illustrate in Figure 2:

- All the leaders $\{\ell_i\}$ induce a clique (black edges).
- Each leader ℓ_i is adjacent to nodes $u_{q,1}$, $\forall q \leq i$, where $q \in \{1, 2, \ldots, N_L\}$ (blue edges).
- For each $i \in \{1, 2, \ldots, N_L\}$ and $j \in \{1, 2, \ldots, D-2\}$, node $u_{i,j}$ is adjacent to nodes $u_{q,j+1}$, $\forall q \leq i$ (red edges).
- For fixed j, all nodes in $\{u_{i,j}\}$ induce a clique, where $i \in \{1, \ldots, N_L\}$ and $j \in \{1, \ldots, D-1\}$ (green edges).

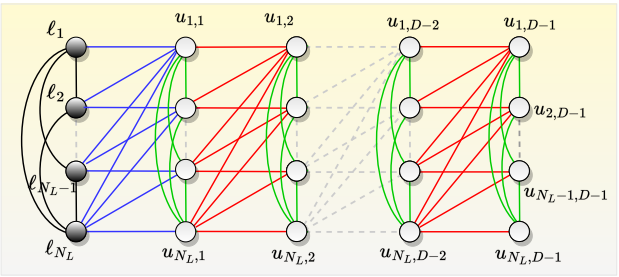


Fig. 2: Construction of graph $\mathcal{G}_1$.

Lemma 3.1. *The set of leader nodes $\mathcal{V}_L = \{\ell_1, \cdots, \ell_{N_L}\}$ in $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ forms a ZFS of $\mathcal{G}_1$.*

Proof. Suppose that all the leader nodes are initially colored black. We observe that in $\mathcal{G}_1$, all leaders, except ℓ_1, have more than one white neighbor in their neighborhoods. So, only ℓ_1 can initiate the zero forcing process. Now, let us examine the adjacent nodes of each leader ℓ_i. We see that ℓ_i is adjacent to exactly i white nodes, out of which $i-1$ are also adjacent to ℓ_{i-1}, for all i. The leader ℓ_1 starts the zero forcing process by coloring its only white neighbor, $u_{1,1}$. As a result, ℓ_2 is left with only one white neighbor, $u_{2,1}$, and thus ℓ_2 colors it. Similarly, ℓ_3 is now left with only one white neighbor that is $u_{3,1}$ because the other two white neighbors $u_{2,1}$ and $u_{1,1}$, which it had in common with ℓ_2 and ℓ_1, respectively, are now colored. This continues until all the nodes in $u_{i,1}$ are colored. Note that there exists a complete graph between all the followers in $u_{i,1}$, therefore, $u_{1,1}$ will only be able to color $u_{1,2}$ once all the other nodes in $u_{i,1}$ are colored. We also know that the nodes in $u_{i,1}$ and in $u_{i,2}$ have similar connections between them as ℓ_i and $u_{i,1}$. So, it follows that the process continues until all the nodes are colored, implying that the given leader set is a ZFS of $\mathcal{G}_1$, which is the desired claim. ∎

Since the leader set $\mathcal{V}_L$ is a ZFS of $\mathcal{G}_1$ (Lemma 3.1), it is strong structurally controllable with $\mathcal{V}_L$ (Theorem 2.1). Next, we demonstrate the robustness of $\mathcal{G}_1$ by showing that its edge set is maximal–any further edge augmentation would compromise its strong structural controllability.

Lemma 3.2. *Consider a graph $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ with a set of leader nodes $\mathcal{V}_L = \{\ell_1, \cdots, \ell_{N_L}\}$. The edge set $\mathcal{E}_1$ is maximal, meaning that adding any extra edge to $\mathcal{G}_1$ would cause $\mathcal{V}_L$ to no longer be a zero forcing set (ZFS) of the modified graph.*

Proof. We begin by noting that $\mathcal{G}_1$ has a unique zero forcing process (ZFP) with $\mathcal{V}_L$ initially colored as black nodes. It means that at each step of the ZFP, there is a unique black node that applies the color-changing rule and colors its only white neighbor.

In particular, in the first step of the ZFP, only ℓ_1 has a unique white neighbor $u_{1,1}$, and therefore, ℓ_1 changes the color of $u_{1,1}$ to black. In the i^{th} step of the ZFP, where $i \leq N_L$, only ℓ_i has a unique white neighbor, denoted as $u_{i,1}$. Consequently, ℓ_i changes the color of $u_{i,1}$ to black. Similarly, in the $(kN_L + j)^{\text{th}}$ step of the ZFP, where $k \in \{1, 2, \cdots, D-2\}$ and $j \in \{1, 2, \cdots, N_L\}$, only the node $u_{j,k}$ has exactly one white neighbor, namely $u_{j,k+1}$, and thus, $u_{j,k}$ changes the color of $u_{j,k+1}$ to black. Now, we proceed to show that the edge set of $\mathcal{G}_1$ is maximal by considering two cases:

(a) Adding an edge between some leader ℓ_i and some follower $u_{j,k}$: Since there is a unique ZFP, in the i^{th} step of the ZFP, ℓ_i will no longer have a unique white neighbor. However, at this point, no other black node in the graph will have a unique white neighbor either. As a result, the coloring of nodes will stop, implying that $\mathcal{V}_L$ is not a ZFS of the modified graph.

(b) Adding an edge between two follower nodes, $u_{j,k}$ and $u_{s,t}$, where $k < t$: Without this new edge, $u_{j,k}$ would be the node with exactly one white neighbor in the $(kN_L + j)^{\text{th}}$ step of the ZFP. However, due to the addition of this extra edge and the uniqueness of the ZFP, no black node in the $(kN_L + j)^{\text{th}}$ step of the ZFP will have a single white neighbor.

Consequently, the ZFP will stop while white nodes are still in the graph, implying that $\mathcal{V}_L$ is not a ZFS of this graph. ■

The number of edges in $\mathcal{G}_1$ is,

$$|\mathcal{E}_1| = N_L \times \left(N - \frac{(N_L+1)}{2}\right). \tag{5}$$

We note that network design 1 utilizes the zero-forcing method to generate a robust and strong structurally controllable graph $\mathcal{G}_1$. Interestingly, we find that $\mathcal{G}_1$ is isomorphic to graph introduced in [8, Section V], which employed a distance-based approach for SSC. Also, from [8, Proposition 5.5], we deduce the following:

Proposition 3.3. *Given positive integers D and N_L, the graph $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ with $N = DN_L$ nodes and N_L leaders is D.*

In $\mathcal{G}_1$, the diameter is $\frac{N}{N_L}$, varying with the total number of nodes N and the number of leaders N_L. However, an intriguing question arises: *Is it possible to design strong structurally controllable graphs with a fixed diameter D, irrespective of the number of nodes N and the number of leaders N_L*. Having graphs with a fixed diameter offers compelling advantages in various scenarios. Firstly, it ensures consistent performance and behavior across network sizes and leadership structures, simplifying the graph design process. Secondly, a fixed diameter, especially of a smaller value, enables the construction of dense networks that enhance communication efficiency, resilience, and overall network robustness. In the subsequent subsection, we provide a solution by designing a strong structurally controllable graph $\mathcal{G}_2$ that is also maximally robust and has a fixed diameter of $D = 2$, with N nodes and $N_L \geq 2$ leaders.

3.2 Network Design 2

Here, we present a design that constructs a graph $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ of a fixed diameter $D = 2$ with a maximal edge set, for given values of N and N_L. Additionally, $\mathcal{G}_2$ is strong structurally controllable with the specified leader set. The number of followers, denoted by N_F, is $N - N_L$. In contrast to $\mathcal{G}_1$, this design ensures that the maximum distance between any two nodes is limited to 2. Next, we explain the construction of $\mathcal{G}_2$ with the following vertex set:

$$\mathcal{V}_2 = \{\ell_i\} \cup \{u_j\},$$

where $i \in \{1, 2, \ldots, N_L\}$ and $j \in \{1, 2, \ldots, N_F\}$. Vertices labeled $\{\ell_1, \ell_2, \ldots, \ell_{N_L}\}$ are leaders and $\{u_1, u_2, \ldots, u_{N_F}\}$ are followers. We adjoin the vertices as follows, and illustrate the construction in Figure 3.

- All the leaders $\{\ell_i\}$ induce a clique among them (black edges).
- Leader ℓ_1 and followers $\{u_1, u_2, \ldots, u_{N_F}\}$ are connected through a path graph starting from ℓ_1 (red edges).
- All the leaders ℓ_i are connected with all the nodes in u_j, where $i \in \{2, \ldots, N_L\}$ and $j \in \{1, \ldots, N_F\}$ (blue edges).

Lemma 3.4. *The set of leader nodes $\mathcal{V}_L = \{\ell_1, \cdots, \ell_{N_L}\}$ in $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ forms a ZFS of $\mathcal{G}_2$.*

Proof. Suppose that all the leader nodes are colored black initially. From the construction of $\mathcal{G}_2$, we observe that all leaders, except ℓ_1, are pair-wise adjacent to all the followers, u_j, $\forall j$. This means that except ℓ_1, all leaders will have more than one white neighbor initially. Since ℓ_1 has only one white neighbor u_1, it will start the zero forcing process by coloring u_1. Next, the rest of the leaders will still have multiple white neighbors in their neighborhoods. However, u_1 has only one white neighbor u_2. Thus, u_1 colors u_2. Subsequently, u_2 will also have only one white neighbor u_3. This stands true for rest of the follower nodes in u_j, where $1 \leq j \leq N_F$. We note that as a result of this unique zero forcing process, the entire graph gets colored, implying that the leader set is a ZFS of $\mathcal{G}_2$. ■

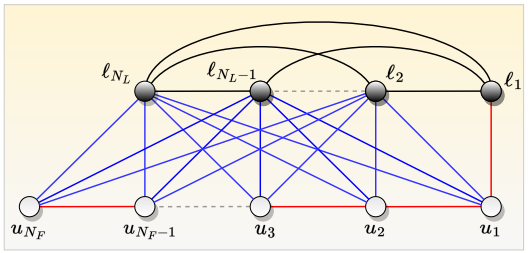


Fig. 3: Graph $\mathcal{G}_2$ with the maximal edge set.

Lemma 3.5. *Let $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ be a graph with a given number of nodes N, and $\mathcal{V}_L = \{\ell_1, \ell_2, \ldots, \ell_{N_L}\}$ is the leader set in $\mathcal{G}_2$. Then, $\mathcal{G}_2$ has the maximal edge set, i.e., the addition of any edge to $\mathcal{G}_2$ will result in the leader set no longer being a ZFS of $\mathcal{G}_2$.*

Proof. $\mathcal{G}_2$ has a unique zero forcing process with $\mathcal{V}_L$ being the set of initially colored black nodes. In the first step of the ZFP, only ℓ_1 has a unique white neighbor u_1. Thus, ℓ_1 colors u_1 to black. In the i^{th} step, where $2 \leq i \leq N_F$, only u_{i-1} has a unique white neighbor u_i and therefore, u_{i-1} changes the color of u_i to black. Now, we can add only two types of edges to $\mathcal{G}_2$, as stated below.

(a) Edge between leader ℓ_1 and follower u_i, where $i \neq 1$. Adding this edge will prevent the initiation of the ZFP, as none of the nodes will have a single white neighbor.

(b) Edge between two followers, i.e., (u_i, u_j), where $i < j$ and $j \neq i+1$. In this case, no black node will have a unique white neighbor in the $(i+1)^{\text{th}}$ step of the ZFP. Thus, the coloring of nodes will stop before all the nodes are colored black, implying that $\mathcal{V}_L$ is not a ZFS of the modified graph.

Hence, $\mathcal{G}_2$ has the maximal edge set, and the addition of any edge to $\mathcal{G}_2$ will result in $\mathcal{V}_L$ no longer being a ZFS of the modified graph. ■

The graphs $\mathcal{G}_1$ (in the previous subsection) and $\mathcal{G}_2$ produce the same number of edges, as shown in (5), when the parameters N and N_L are identical. However, unlike $\mathcal{G}_1$, the diameter of $\mathcal{G}_2$ is fixed. In $\mathcal{G}_2$, all leaders induce a clique, thus $d(\ell_i, \ell_j) = 1$. Additionally, each leader ℓ_i, where $i > 1$, is adjacent to all the follower nodes, thus, $d(\ell_i, u_j) = 1$. Moreover, ℓ_1 is adjacent to u_1. From these observations, it is obvious that $D = 2$.

Proposition 3.6. *The diameter of graph $\mathcal{G}_2$ with N nodes and N_L leaders is 2.*

It is worth mentioning that the graph $\mathcal{G}_2$ exhibits *optimal robustness* for specific values of N_L, such as $N_L = N-2$. This implies that among all graphs with N nodes and diameter

$D = 2$, $\mathcal{G}_2$ has the minimum Kirchhoff index K_f (hence, maximum robustness). The optimality arises because, for $N_L = N - 2$ and $D = 2$, $\mathcal{G}_2$ belongs to a particular class of graphs known as *optimal clique chains*, which have the maximum robustness among all graphs with N nodes and diameter D [3]. Consequently, given N, $N_L = N - 2$, and $D = 2$, $\mathcal{G}_2$ is a strong structurally controllable graph with the maximum robustness. Figure 4 illustrates such a graph.

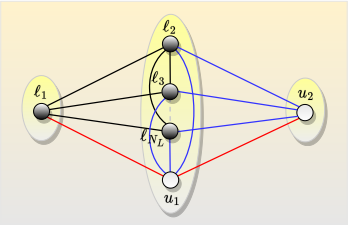


Fig. 4: For $N_L = N - 2$, $\mathcal{G}_2$ is an optimal clique chain with N nodes and diameter $D = 2$.

The diameter of $\mathcal{G}_1$ is determined by N_L and N, whereas $\mathcal{G}_2$ has a constant diameter. However, we now consider an intriguing question: Can we make the graph's diameter a design input alongside N and N_L? Allowing flexibility in diameter selection could prove invaluable in different use cases where optimizing specific performance metrics or accommodating network constraints, such as communication times, require different diameters. To address this, in the next section, we present graph constructions that also take the diameter D as an input and generate controllable and robust networks.

4 Constructing Controllable and Robust Networks for Given N, N_L, and D

In this section, we present more flexible graph constructions incorporating three design parameters: the number of nodes N, the number of leaders N_L, and the diameter D. In particular, we consider the following problem:

Problem 2: How can we design graphs that are both strong structurally controllable and maximally robust for given inputs, including N, N_L and D?

For this, we initially present a design that combines network designs 1 and 2 in the previous section and produces graphs for given N, N_L, and $2 \leq D < N/N_L$. Subsequently, in subsection 4.2, we present a general design capable of producing strong structurally controllable and robust graphs while affording a more comprehensive range of graph diameters.

4.1 Network Design 3 (Combining Designs 1 and 2)

This subsection presents a design for graph $\mathcal{G}_3 = (\mathcal{V}_3, \mathcal{E}_3)$, which is obtained by combining two previous designs, namely $\mathcal{G}_1$ and $\mathcal{G}_2$. Similar to $\mathcal{G}_1$ and $\mathcal{G}_2$, the combined graph $\mathcal{G}_3$ demonstrates strong structural controllability and maximal robustness. To combine graphs, we first introduce a *graph gluing* operation, defined below.

Definition 4.1. (Graph Gluing) *Consider two graphs $H_1 = (V_1, E_1)$ and $H_2 = (V_2, E_2)$, along with a pair of subsets $S_1 = \{x_1, \cdots, x_m\} \subseteq V_1$ and $S_2 = \{y_1, \cdots, y_m\} \subseteq V_2$. The graphs H_1 and H_2 are glued through S_1 and S_2 to obtain a new graph H by identifying vertices $x_i \in S_1$ and $y_i \in S_2$, for each $i \in \{1, \cdots, m\}$. The identification of x_i and y_i results in a new vertex w_i whose neighborhood is the union of the neighborhoods of x_i and y_i, encompassing all adjacent vertices of x_i in H_1 and y_i in H_2. A glued graph is denoted by $H = H_1 \oplus H_2$.*

Figure 5 illustrates gluing of graphs H_1 and H_2 through $S_1 = \{x_1, x_2\}$ and $S_2 = \{y_1, y_2\}$ to obtain a glued graph $H = H_1 \oplus H_2$.

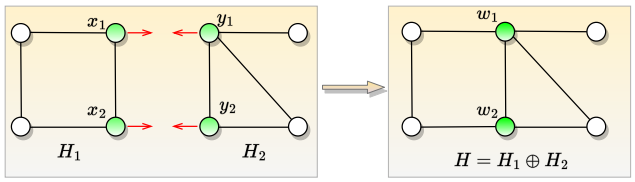


Fig. 5: The graph H obtained by gluing graphs H_1 and H_2 through S_1 and S_2 (green nodes).

Next, for given N_L, N and D, where $2 \leq D < \frac{N}{N_L}$ and $N \geq DN_L + 1$, we construct $\mathcal{G}_3$ as following.

- First, construct a graph $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$, with leader set $\mathcal{V}_L^1 = \{\ell_1^1, \cdots, \ell_{N_L}^1\} \subseteq \mathcal{V}_1$, where $|\mathcal{V}_L^1| = N_L$ and $|\mathcal{V}_1| = N_L \times (D-1)$.
- Second, construct $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$, with leader set $\mathcal{V}_L^2 = \{\ell_1^2, \cdots, \ell_{N_L}^2\} \subseteq \mathcal{V}_2$, where $|\mathcal{V}_2| = N - (|\mathcal{V}_1| - N_L)$.
- Now, consider a pair of subsets, $S_1 = \{u_{1,D-2}, \cdots, u_{N_L,D-2}\} \subseteq \mathcal{V}_1$ and $S_2 = \{\ell_1^2, \cdots, \ell_{N_L}^2\} \subseteq \mathcal{V}_2$.
- Finally, glue $\mathcal{G}_1$ and $\mathcal{G}_2$ through S_1 and S_2, as in Definition 4.1, to obtain a glued graph $\mathcal{G}_3 = \mathcal{G}_1 \oplus \mathcal{G}_2$ with the leader set $\mathcal{V}_L = \mathcal{V}_L^1$. Here $|\mathcal{V}_3| = N$ and $|\mathcal{V}_L| = N_L$.

Figure 6 illustrates an example of $\mathcal{G}_3$ for $N_L = 3$, $N = 15$, and $D = 4$. We first construct $\mathcal{G}_1$ and $\mathcal{G}_2$, each with nine nodes. Then, we glue them together through $S_1 = \{u_{1,2}, u_{2,2}, u_{3,2}\} \subset \mathcal{V}_1$ and $S_2 = \{\ell_1^2, \ell_2^2, \ell_3^2\} \subset \mathcal{V}_2$, as shown in Figure 6.

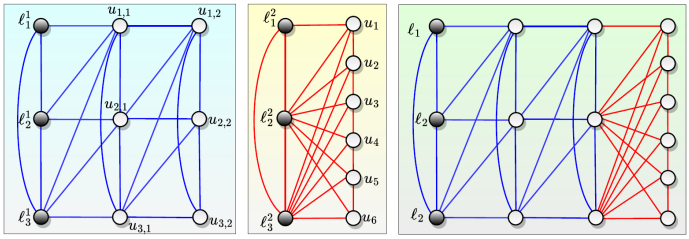


Fig. 6: Gluing $\mathcal{G}_1$ and $\mathcal{G}_2$ to get $\mathcal{G}_3$ with $N = 15$, $N_L = 3$, and $D = 4$.

Next, we show that V_L is a ZFS of $\mathcal{G}_3$, thereby making $\mathcal{G}_3$ strong structurally controllable.

Lemma 4.1. *The leader set $\mathcal{V}_L = \{\ell_1, \cdots, \ell_{N_L}\}$ in $\mathcal{G}_3 = (\mathcal{V}_3, \mathcal{E}_3)$ is a ZFS of $\mathcal{G}_3$.*

Proof. Suppose $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ be the two graphs that are glued together through subsets $S_1 \subset \mathcal{V}_1$ and

$S_2 \subset \mathcal{V}_2$ to obtain $\mathcal{G}_3$ (as explained in $\mathcal{G}_3$'s construction). Since the leader set $\mathcal{V}_L$ of $\mathcal{G}_3$ is also the leader set of $\mathcal{G}_1$ and the neighborhoods of nodes in $\mathcal{V}_1 \setminus S_1$ do not change due to gluing, all the nodes in $\mathcal{V}_1$ get colored due to the ZFP (by Lemma 3.1). Moreover, this ZFS is unique, and nodes in S_1 are the last to get colored. Crucially, none of the nodes in S_1 color any other node in $\mathcal{V}_1$. Therefore, if any new nodes become adjacent to S_1, they will still be colored black due to the ZFS. Next, due to gluing, nodes in S_1 are identified with nodes in S_2, which are also the leaders of $\mathcal{G}_2$. By Lemma 3.4, all nodes in $\mathcal{G}_2$ get colored. Consequently, all nodes in $\mathcal{G}_3$ are eventually colored. Hence, $\mathcal{V}_L$ is a ZFS of $\mathcal{G}_3$. ■

Next, we observe that the leader set $\mathcal{V}_L$ induces a unique ZFP in $\mathcal{G}_3$. By employing similar arguments as in Lemmas 3.2 and 3.5, we can demonstrate that the edge set in $\mathcal{G}_3$ is maximal. In other words, we have the following.

Proposition 4.2. *The edge set in $\mathcal{G}_3 = (\mathcal{V}_3, \mathcal{E}_3)$ is maximal with the given leader set $\mathcal{V}_L$ (as defined in $\mathcal{G}_3$'s construction). In other words, the addition of any edge to $\mathcal{G}_3$ will cause $\mathcal{V}_L \subset \mathcal{V}_3$ to no longer be a ZFS of $\mathcal{G}_3$.*

The three designs in Sections 3.1, 3.2, and 4.1 yield graphs demonstrating strong structural controllability and maximal robustness for the given input parameters. We discussed that, for given N and N_L, it is possible to construct a graph with a diameter D in the range of 2 to N/N_L. In particular, $\mathcal{G}_1$ and $\mathcal{G}_2$ construct graphs with fixed diameters of 2 and N/N_L, respectively. However, $\mathcal{G}_3$ affords some flexibility regarding D, albeit with certain limitations. Notably, the permissible range of D diminishes as N_L increases. Also, since $\mathcal{G}_3$ combines two designs, $\mathcal{G}_1$ and $\mathcal{G}_2$, it does not offer a general solution. Consequently, $\mathcal{G}_3$ offers a relatively restricted scope of achievable diameters. Moreover, the robustness of $\mathcal{G}_3$ remains confined by that of its components, $\mathcal{G}_1$ and $\mathcal{G}_2$. This leads us to the question: *Can we devise a more general approach to design graphs for given N and N_L, allowing for a wider range of diameters D and potentially achieving higher robustness compared to the previous graphs?*

4.2 Network Design 4 – A Comprehensive Solution

This subsection presents a novel approach for generating graph $\mathcal{G}_4 = (\mathcal{V}_4, \mathcal{E}_4)$, applicable to any combination of N, N_L, and D where $D \leq N_L$. $\mathcal{G}_4$ is strong structurally controllable and maximally robust. Our construction leverages the concept of clique chains, which represent graphs with the highest possible robustness—indicated by the minimum Kirchhoff index—among all graphs comprising N nodes and a diameter D. We also compare the robustness of $\mathcal{G}_4$ with optimal clique chains with the same parameters in subsection 4.3. We begin by defining the clique chain below.

Definition 4.2. *(*Clique Chain*) Consider a set of positive integers $n_1, \cdots n_{D+1}$ and suppose $N = \sum_{i=1}^{D+1} n_i$. Also, consider a path graph $\mathcal{P}_{D+1}$ of diameter D. Then clique chain is a graph with N nodes and diameter D that is obtained by replacing each node i in $\mathcal{P}_{D+1}$ with a clique of size n_i (i.e., $\mathcal{K}_{n_i}$) such that the vertices in distinct cliques are adjacent if and only if the corresponding original vertices in the path graph are adjacent.*

We denote a clique chain with diameter D by $\mathcal{C}_D(n_1, \cdots, n_{D+1})$, or simply $\mathcal{C}_D$ if the context is obvious. Figure 7 illustrates an example of clique chain.

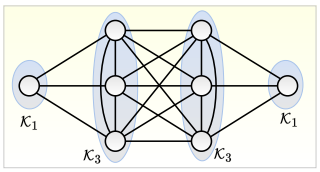


Fig. 7: A clique chain $\mathcal{C}_3(1, 3, 3, 1)$ of diameter $D = 3$.

The construction of $\mathcal{G}_4$ relies on the partitioning of an integer into D parts, which entails expressing a positive integer P as the sum of D positive integers: $P_1, P_2, \ldots, P_D$, such that $\sum_{i=1}^{D} P_i = P$. For instance, consider the positive integer $P = 6$. We can partition it into four parts as $P_1 = P_2 = 1$ and $P_3 = P_4 = 2$. Constructing $\mathcal{G}_4$ involves partitioning the integers N_L and $(N - N_L)$—representing the number of leaders and followers, respectively—into D parts in a specific manner. The Appendix provides this partitioning scheme. Next, we present the construction of $\mathcal{G}_4$.

Construction of $\mathcal{G}_4$: For given N, N_L, and $D \leq N_L$, we proceed as follows. Figure 8 shows the construction of $\mathcal{G}_4$.

- Partition the number of leaders N_L into D parts (using the method described in the Appendix). Let this partition be $L_1, \cdots, L_D$, where each L_i is a positive integer. Note that $\sum_{i=1}^{D} L_i = N_L$.
- Similarly, partition the number of followers, i.e., $N - N_L$, into D parts. Assume this partition is $F_1, \cdots, F_D$.
- Distribute the leaders into D cells, where the i^{th} leader cell contains L_i leaders. We denote the leaders in the i^{th} leader cell by $\ell_{j,i}$, where $j \in \{1, 2, \cdots, L_i\}$. Similarly, arrange the followers in D cells while assigning F_i followers to the i^{th} cell. The follower nodes in the i^{th} follower cell are denoted by $u_{j,i}$, where $j \in \{1, 2, \cdots, F_i\}$.
- The leaders induce a clique chain, $\mathcal{C}_{D-1}(L_1, \cdots, L_D)$ (as in Definition 4.2).
- The leader $\ell_{1,1}$ and the follower nodes induce the following path (red edges in Figures 8 and 9):

$$< \ell_{1,1}, u_{1,1}, \cdots, u_{F_1,1}, u_{1,2}, \cdots, u_{F_2,2}, \cdots, u_{1,F_D}, \cdots, u_{F_D,F_D} > .$$

- For each $i > 1$, all the leaders in the i^{th} leader cell are pair-wise adjacent to all the followers in i, $i-1$, and $i-2$ follower cells containing F_i, F_{i-1}, and F_{i-2} nodes, respectively (blue edges in Figures 8 and 9).
- Additionally, for each $i > 1$, the follower node $u_{F_i-1,i}$ in the i^{th} follower cell is pair-wise adjacent to all the follower nodes in the $i+1$ follower cell (green edges in Figures 8 and 9).

The partitioning of N_L into $L_1, \cdots, L_D$ (as described in Appendix) is such that $L_1 = 1$. Similarly, $N - N_L$ is partitioned into $F_1, \cdots, F_D$, such that $F_D = 1$.

Figure 9 provides an example of $\mathcal{G}_4$ for $N = 16$, $N_L = 6$ and $D = 4$. Here, the six leaders are partitioned into $D = 4$ cells containing $L_1 = 1$, $L_2 = L_3 = 2$, and $L_4 = 1$ leaders. Similarly, ten followers are partitioned into $D = 4$ cells containing $F_1 = F_2 = F_3 = 3$ and $F_4 = 1$ followers. The edges are established based on the construction principles

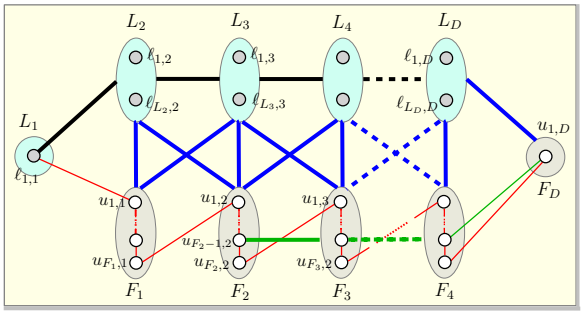


Fig. 8: Construction of graph $\mathcal{G}_4$. Leader cells are highlighted in blue, and follower cells are depicted in gray. A bold line connecting two cells indicates that each node within one cell is pairwise adjacent to every node in the other cell.

discussed above. Next, we show that the leader set $\mathcal{V}_L$ in $\mathcal{G}_4$ is indeed a ZFS of $\mathcal{G}_4$, thus, rendering the graph strong structurally controllable.

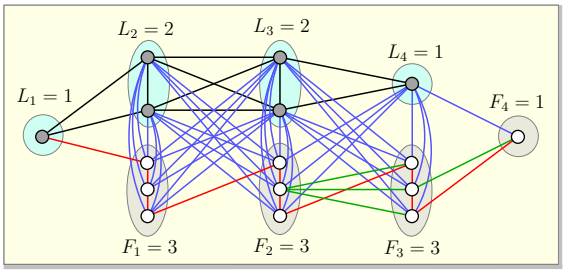


Fig. 9: Graph $\mathcal{G}_4$ with $N = 16$, $N_L = 6$, and $D = 4$.

Lemma 4.3. *For a graph $\mathcal{G}_4 = (\mathcal{V}_4, \mathcal{E}_4)$ with N nodes and N_L leaders, the proposed leader set $\mathcal{V}_L = \{\ell_{1,1}, \ell_{1,2}, \ell_{2,2} \cdots, \ell_{L_2,2}, \cdots, \ell_{1,D}, \ell_{2,D} \cdots, \ell_{L_D,D}, \}$ is a ZFS of $\mathcal{G}_4$.*

Proof. The leaders are divided into D cells, each of size L_i, such that $\sum_{i=1}^{D} L_i = N_L$. Assume that initially, all the leaders are colored black and followers are white. From the construction of $\mathcal{G}_4$, it is evident that for each j, a leader $\ell_{j,i}$, where $i > 1$, is adjacent to followers in multiple follower cells. This means $\ell_{1,1}$ is the only leader with a unique white neighbor $u_{1,1}$. This enables $\ell_{1,1}$ to initiate the ZFP by coloring $u_{1,1}$. Next, $u_{2,1}$ being the only white neighbor of $u_{1,1}$, will be colored. This procedure continues until all the follower nodes in the first follower cell are colored. Now, consider $u_{1,2}$ from the second follower cell, which is the only white neighbor of $u_{F_1,1}$. Thus, $u_{F_1,1}$ will color $u_{1,2}$. Then $u_{1,2}$ will color $u_{2,2}$, and so on until $u_{F_2-1,2}$ is colored. Note that $u_{F_2-1,2}$ is adjacent to all the followers in the third follower cell, meaning that it has multiple white neighbors and hence, cannot color them. However, all the leaders in the second ($i = 2$) leader cell are adjacent to all the followers in the first and second (i.e., i and $i-1$) follower cells. This means that for each leader $\ell_{j,2}$ in the second leader cell, $u_{F_2,2}$ is the only white node in $\ell_{j,2}$'s neighborhood. Thus, $u_{F_2,2}$ will be colored by any such leader. Now, $u_{F_2,2}$ has only $u_{1,3}$, which is in the next (third) follower cell, as its white-colored neighbor. Therefore, $u_{F_2,2}$ will color $u_{1,3}$. The above coloring pattern continues until all the followers are colored, confirming that the leader set $\mathcal{V}_L$ is a ZFS of $\mathcal{G}_4$. ■

Next, we discuss that the diameter of $\mathcal{G}_4$ is indeed D. For this, we consider the *distance partition* of $\mathcal{G}_4$ with respect to $\ell_{1,1}$, which is the partitioning of nodes (leaders and followers) based on their distances with $\ell_{1,1}$. Let $\mathcal{S}_i$ be the set of nodes in $\mathcal{G}_4$ that are at a distance i from $\ell_{1,1}$, i.e.,

$$\mathcal{S}_i = \{x \in \mathcal{V}_4 : \ d(\ell_{1,1}, x) = i\}.$$

We observe that $\mathcal{S}_1$ consists of leaders in the second leader cell and followers in the first leader cell (as Figures 8 and 10 also show). Similarly, $\mathcal{S}_2$ contains leaders in the third leader cell and followers in the second leader cell. In general, $\mathcal{S}_i$ consists of leaders in the $(i+1)^{\text{st}}$ leader cell and followers in the i^{th} follower cell, i.e.,

$$\mathcal{S}_i = \{\ell_{j,i+1}, \ \forall j\} \cup \{u_{k,i}, \ \forall k\}.$$

In particular, $\mathcal{S}_D$ consists of the last follower cell, which only contains $u_{1,D}$, thus showing the distance $d(\ell_{1,1}, u_{1,D}) = D$. It is easy to see that $\ell_{1,1}$ and $u_{1,D}$ are the nodes that are farthest from each other in $\mathcal{G}_4$; hence, the diameter of $\mathcal{G}_4$ is indeed $d(\ell_{1,1}, u_{1,D}) = D$. Figure 10 illustrates this distance partition.

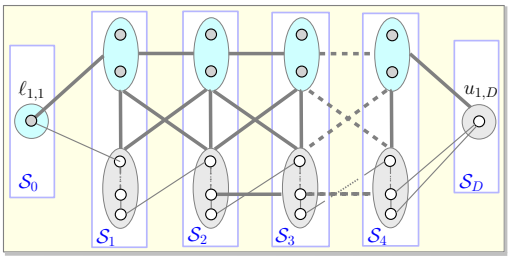


Fig. 10: The distance partition of $\mathcal{G}_4$ with respect to $\ell_{1,1}$. Leader and follower cells are in blue and gray, respectively. $\mathcal{S}_i$ indicates nodes in the i^{th} cell of the distance partition.

We proceed to demonstrate the maximality of the edge set in $\mathcal{G}_4$. In particular, we show the following:

Lemma 4.4. *For given N, N_L and D, graph $\mathcal{G}_4 = (\mathcal{V}_4, \mathcal{E}_4)$ with the leader set $\mathcal{V}_L$ (constructed as above) has maximal edge set, i.e., including an extra edge to $\mathcal{G}_4$ would result in either a reduction in the graph's diameter D and/or the leader set $\mathcal{V}_L$ no longer being a ZFS of the resulting graph.*

Proof. Let's consider the new edge added to $\mathcal{G}_4$. This edge can fall into one of three categories:

(a) *Between two leader nodes:* Assume $\ell_{x,i}$ and $\ell_{y,j}$ are two leaders not initially adjacent in $\mathcal{G}_4$, belonging to the i^{th} and j^{th} leader cells, respectively. Note that $|i-j| > 1$, as otherwise they will already be adjacent in $\mathcal{G}_4$. Consider the distance partition of $\mathcal{G}_4$, as discussed earlier and depicted in Figure 10. We observe that $\ell_{x,i} \in \mathcal{S}_{i-1}$ and $\ell_{y,j} \in \mathcal{S}_{j-1}$. Consequently, the edge between $\ell_{x,i}$ and $\ell_{y,j}$ adjoins a node in $\mathcal{S}_{i-1}$ with a node in $\mathcal{S}_{j-1}$, where $|(i-1)-(j-1)| > 1$. As a result, the distance between $\ell_{1,1}$ and $u_{1,D}$, the furthest nodes in $\mathcal{G}_4$, decreases. Therefore, the diameter of the resulting graph will be smaller than D.

(b) *Between a leader and a follower node:* Consider a new edge between leader $\ell_{x,i}$ from the i^{th} leader cell and follower

$u_{y,j}$ from the j^{th} follower cell. If $(i-j) \in \{0,1,2\}$, then $\ell_{x,i}$ and $u_{y,j}$ are already adjacent in $\mathcal{G}_4$ by construction. So, we assume that either $i-j > 2$ or $i-j < 0$. Now, in the distance partition of $\mathcal{G}_4$ (shown in Figure 10), $\ell_{x,i} \in \mathcal{S}_{i-1}$ and $u_{y,j} \in \mathcal{S}_j$. Thus, the new edge connects a node in $\mathcal{S}_{i-1}$ with a node in $\mathcal{S}_j$, where $|(i-1)-j| > 1$. As a result, the distance between $\ell_{1,1}$ and $u_{1,D}$ decreases, leading to a reduction in the graph's diameter.

(c) *Between two follower nodes:* Initially, consider the case of a new edge between two followers $u_{x,i}$ and $u_{y,i}$ in the same follower cell i. Without loss of generality, let $x < y$. Then $y - x > 1$. In the ZFP of $\mathcal{G}_4$, $u_{x,i}$ is the only node to color $u_{x+1,i}$ (as also discussed in Lemma 4.3). However, an edge between $u_{x,i}$ and $u_{y,i}$ prevents $u_{x,i}$ from coloring $u_{x+1,i}$ due to multiple white neighbors. Consequently, the ZFP will stop before all followers are colored, implying that $\mathcal{V}_L$ is no longer a ZFS of the graph. A similar argument can be applied to show that an edge between two followers, that are not adjacent in $\mathcal{G}_4$, and are in the different follower cells will hinder the ZFP, preventing all the followers from being colored. Hence, including any new edge to $\mathcal{G}_4$ either reduces the graph's diameter or disrupts the property of $\mathcal{V}_L$ being a ZFS for the graph. ■

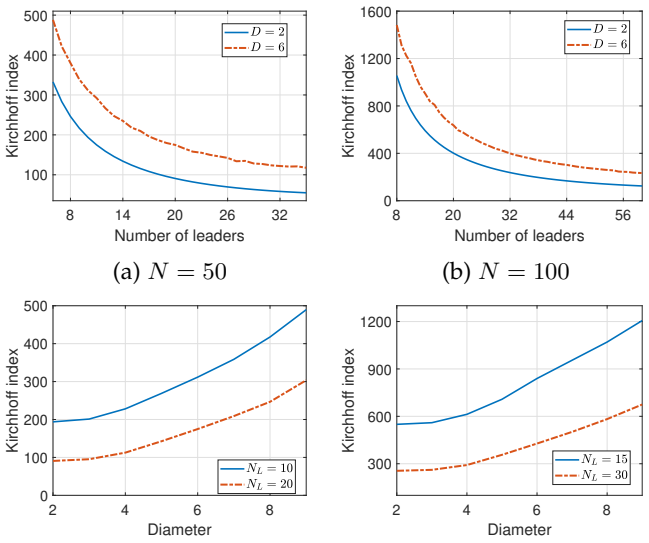
In general, $\mathcal{G}_4$ offers a broader range of D values for network design compared to $\mathcal{G}_3$. However, when $N > (N_L)^2$, $\mathcal{G}_3$ affords more D values for designing networks. In the following subsection, we analyze the robustness of $\mathcal{G}_4$.

4.3 Robustness Analysis and Experimental Evaluation

This subsection presents an experimental evaluation of the robustness of $\mathcal{G}_4$, measured through the Kirchhoff index K_f. Subsequently, the optimality of $\mathcal{G}_4$ for certain combinations of N, N_L, and D is validated.

Figures 11(a) and (b) illustrate the variation of the Kirchhoff index with the number of leaders in $\mathcal{G}_4$ for $N = 50$ and 100. The graph diameters are set at 2 and 6. The results reveal increased graph robustness (indicated by decreasing K_f values) as the number of leaders rises. This is attributed to the additional leaders that enable augmenting more edges within the network while satisfying the strong structural controllability condition, consequently leading to the reduced K_f. The subsequent plots in Figures 11 (c) and (d) illustrate the behavior of K_f with respect to graph diameter D while keeping the number of leaders constant. Intuitively, as the graph diameter increases, the number of edges tends to decrease, thereby diminishing graph robustness. The presented plots affirm this decline in graph robustness as diameter increases.

Next, we discuss the robustness performance of $\mathcal{G}_4$. For given N, N_L, and D, determining a graph that maximizes robustness (minimizes K_f) becomes a computational challenge. Even for given N and D, finding a graph with the minimum K_f is computationally intractable. In this context, we know that such graphs are clique chains of the form $\mathcal{C}_D(n_1, n_2, \cdots, n_{D+1})$, where $\sum_{i=1}^{D+1} n_i = N$ [3, Corollary 3.2]. However, there is no general way of computing an optimal distribution of n_i's minimizing K_f for given N and D. In [3, Table 1], for select cases of N and D, optimal clique chains (minimizing K_f for given N and D) are provided through exhaustive search. Our challenge goes beyond this

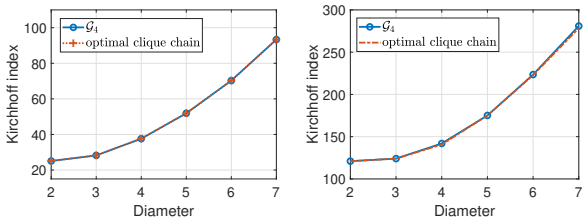


(a) $N = 50$ (b) $N = 100$

(c) $N = 50$ (d) $N = 100$

Fig. 11: (a) and (b) Kirchhoff index of $\mathcal{G}_4$ as function of N for fixed D. (c) and (d) K_f as function of D for fixed N_L.

as we aim to ensure that the graphs have minimum K_f and are strong structurally controllable. This is challenging, especially because finding the minimum number of leaders rendering the graph strong and structurally controllable is NP-hard [38], [39]. For this, we utilize [10, Proposition 4.8] stating that $N_L = N-D$ leaders are needed to make a clique chain with N nodes and D diameter strong structurally controllable. Based on this, we know optimum graphs with the minimum K_f for particular instances of N, D, and N_L, and we compare their robustness with $\mathcal{G}_4$ for the same design parameters. Figure 12 plots the results. Remarkably, $\mathcal{G}_4$ consistently exhibits global optimum robustness across nearly all cases. In other cases, it produces nearly optimum graphs. The alignment between the robustness of optimal clique chains and that of $\mathcal{G}_4$ substantiates the efficacy of the proposed design.



(a) $N = 26$ (b) $N = 122$

Fig. 12: Robustness comparison of $\mathcal{G}_4$ and optimal clique chains.

We further analyze the robustness of $\mathcal{G}_4$ for a range of values of N, N_L, and D. For this, we conducted experiments using a database of connected isomorphic graphs presented in [40] encompassing graphs with up to $N = 10$ nodes. For each $N \in \{8, 9, 10\}$, we compute graphs with specified diameters D (as in Table 1). Subsequently, for every pairing of N and D, we specify N_L, and compute all graphs that are strong structurally controllable with N_L leaders. In other

N	D	N_L	$K_f(\mathcal{G}_4)$	Optimum K_f	Graphs with N, N_L, D
8	2	2	21.2440	21.2440	3 (0.03%)
		3	13.1897	13.1897	424 (3.81%)
		4	9.8240	9.8054	2678 (24.09%)
		5	8.1569	8.1569	987 (8.88%)
		6	7.3333	7.3333	61 (0.55%)
	3	3	15.0371	13.3386	2820 (25.37%)
		4	12.3260	11.6104	2692 (24.22%)
		5	11.0159	11.0159	196 (1.76%)
	4	4	19.0333	19.0333	203 (1.83%)
9	2	2	27.7994	27.7994	3 (0.001%)
		3	17.1865	17.1865	1152 (0.44%)
		4	12.6409	12.5781	36693 (14.05%)
		5	10.2675	10.2445	48666 (18.64%)
		6	8.9666	8.9666	4902 (1.88%)
		7	8.2857	8.2857	102 (0.04%)
	3	3	19.2594	17.3772	26832 (10.28%)
		4	15.4066	13.6179	97227 (37.24%)
		5	12.4155	12.4155	23548 (9.02%)
		6	11.9583	11.9583	491 (0.19%)
	4	4	20.7130	20.7130	7702 (2.95%)
		5	19.5714	19.5714	730 (0.28%)
10	4	4	22.6729	22.4179	362487 (3.09%)
		5	21.6283	21.4458	73284 (0.625%)
		6	20.6905	20.6905	2291 (0.02%)
	5	5	33.0833	33.0833	1882 (0.016%)

TABLE 1: Experimental results for the robustness of $\mathcal{G}_4$. Highlighted rows indicate cases for which $\mathcal{G}_4$ exhibit the maximum robustness.

words, we identify all graphs with N nodes and D diameter whose minimum zero forcing set size is N_L. Proceeding further, for each combination of N, N_L, and D, we compute graphs with the minimum K_f (maximum robustness) through exhaustive search. The optimum values of K_f are presented in the second-to-last column. Moreover, the last column shows the total count of graphs conforming to these design parameters, accompanied by the percentage of connected graphs with N nodes satisfying these parameters. Next, we generate instances of $\mathcal{G}_4$ for the designated N, N_L, and D values and calculate their respective K_f values. The highlighted rows within the table signify scenarios where $\mathcal{G}_4$ exhibit the minimum K_f, and therefore, globally optimum robustness. For instance, the last row of the table shows that there are 1882 graphs accounting for 0.016% of the total of 12005168 connected graphs with $N = 10$ nodes. All of these 1882 graphs have a diameter of $D = 5$ and are strong structurally controllable with $N_L = 5$ leaders. Out of these graphs, $\mathcal{G}_4$ has the maximum robustness. Our observations consistently reveal $\mathcal{G}_4$ as the optimal choice for a wide range of specified parameters. This is significant, particularly given the computational complexities associated with this problem. Moreover, even in a less restrictive setting (only considering N and D), there are no available techniques to obtain globally optimal solution based on the design criteria explained earlier.

5 Distributed construction using graph grammars

This section extends the theoretical design proposed in Section 4.2 by addressing the following issue.

Problem 3: How can we implement network $\mathcal{G}_4$ in a distributed and scalable manner?

To tackle this challenge, we harness the concept of *graph grammars*, a distributed approach to graph construction [16], [41]–[44]. In this distributed methodology, the graph construction is driven by *local* rules governing the interactions among individual agents. Here, the agents are individual nodes of the graph. Importantly, these agents make decisions about edge augmentation based solely on local information without relying on any global knowledge of the network. For this, graph grammars provide a framework for designing local rules enabling agents to collaboratively generate the desired network structure, which in our case is the construction of the graph $\mathcal{G}_4$. The main idea entails assigning agents 'labels' that are regularly updated. Rules are then defined for the edge augmentation among agents with various labels. These rules are essentially derived from the construction of graph $\mathcal{G}_4$, as proposed in Section 4.2. This distributed approach is practical, scalable, and well-aligned with the theoretical framework developed in Section 4.2.

Each rule consists of a 'left-hand-side' (input) subgraph ($\mathcal{G}_L$), which must exist within the graph $\mathcal{G}$ for the rule to be applicable. Additionally, there is a 'right-hand-side' (output) subgraph ($\mathcal{G}_R$) that defines the changes in the graph following the successful application of rules [16], [41]. In essence, when a subgraph isomorphic to $\mathcal{G}_L$ is found in the graph $\mathcal{G}$, the corresponding rule is executed. This execution results in the creation of the associated subgraph $\mathcal{G}_R$, thereby modifying $\mathcal{G}$ into $\mathcal{G}'$. To initialize the process, each node is assigned an initial set of labels. A set of rules, denoted by $\mathcal{R} = \{r_0, r_1, \cdots, r_n\}$, is defined to modify the connections between nodes and update their labels. These rules encompass a range of actions:

- *Constructive:* These rules create edges between nodes that did not have edges in $\mathcal{G}_L$, i.e., $\mathcal{G}_L$ is a subgraph of $\mathcal{G}_R$, denoted by $\mathcal{G}_L \subset \mathcal{G}_R$.
- *Destructive:* These rules remove edges between nodes that exist in $\mathcal{G}_L$, i.e., $\mathcal{G}_R \subset \mathcal{G}_L$.
- *Label changing rules:* These rules modify the labels of one or more involved nodes.
- *Mixed rules:* Most rules entail both a change in labels and the creation or deletion of edges.

To illustrate the design process of graph grammars, we consider the following simple example. Consider an initial graph $\mathcal{G}$ with an empty edge set, as Figure 13(a) illustrates. The goal is to design graph grammars to transform $\mathcal{G}$ into $\mathcal{G}'$, a configuration composed of star graphs $\mathcal{K}_{1,4}$, each featuring a central node adjacent to four leaf nodes. Initially, all nodes are labeled a, as shown in Figure 13(a). The rules designed to change the labels and the edge set to obtain the desired graph $\mathcal{G}'$ are as follows:

$$(r_0) \quad a \quad a \quad \rightharpoonup \quad \ell_1 \text{—} c\,,$$
$$(r_1) \quad \ell_i \quad a \quad \rightharpoonup \quad \ell_{i+1} \text{—} c\,, \qquad 1 \le i \le 3.$$

Upon applying rule r_0, two non-adjacent nodes labeled as a become adjacent. Simultaneously, their labels are updated; one is designated ℓ_1, while the other gets the label c. In a similar vein, as a result of rule r_1, a node with a label ℓ_i, where $i \le 3$, is adjoined with a node labeled a, thereby changing their labels to ℓ_{i+1} and c, respectively. Notably, these rules induce modifications in the edge set and the node labels. A possible sequence of transformations

is illustrated in Figure 13, illustrating the emergence of multiple instances of $\mathcal{K}_{1,4}$.

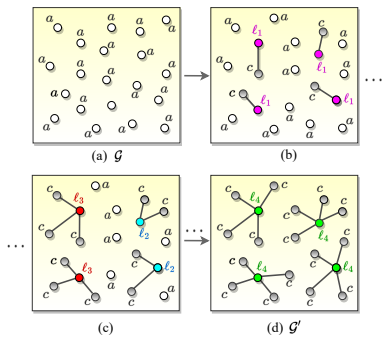


Fig. 13: Application of graph grammars to form stars $\mathcal{K}_{1,4}$.

Next, we discuss graph grammars to construct the graph $\mathcal{G}_4$ (as discussed in Subsection 4.2). A set of rules $\mathcal{R}$ generating graph $\mathcal{G}_4$ for a given number of nodes N, leaders N_L, and diameter D is shown in Table 2. We note that all the nodes are initially labeled a, except one node, which is labeled $S_{1,1}$. This node is picked randomly and acts as a seed node for our graph. Then, the rules are defined such that the required graph $\mathcal{G}_4$, *"grows"* from that seed node. Hence, the determination of leaders and followers within the graph relies on the application of rules. We classify the rules $\mathcal{R}$ into four parts that work coherently to form $\mathcal{G}_4$ with the given specification.

- $\Pi_1 \rightarrow$ Rules creating two connected path graphs of length $D-1$; one for each leader and follower chains.
- $\Pi_2 \rightarrow$ Rules expanding each node in the chains by creating a path graph with the actual number of nodes in each partition.
- $\Pi_3 \rightarrow$ Rules modifying connections between the follower partition to accommodate the desired formation of a path graph between the followers and the first leader.
- $\Pi_4 \rightarrow$ Rules maximizing the number of edges between and within all the partitions.

An example of graph grammars constructing $\mathcal{G}_4$ with number of nodes $N = 13$, number of leaders $N_L = 5$, and diameter $D = 4$ is illustrated in Figure 14. Here, all the nodes are initially labeled as a except for one node, which is labeled as $S_{1,1}$. The graph originates from the seed node $S_{1,1}$, and the rules applied to construct the graph are color-coded to distinguish the edges.

6 Conclusion

The paper introduced multiple approaches for constructing network graphs that balance network controllability and robustness. Our methods leverage the concept of zero forcing to maximize robustness by strategically adding edges while maintaining SSC. We provided various network designs tailored to specific parameters: the total number of nodes N, the number of leaders N_L, and the network diameter D. Notably, all designs demonstrated SSC within the specified parameters. Designing networks that are strong structurally controllable and highly robust while adhering to various structural constraints is a computationally intensive challenge. Our solutions, however, generate networks with the desired specifications in polynomial time ensuring scalability while achieving near-optimal robustness. We also demonstrated a distributed approach to constructing these graphs, relying on graph grammars. These grammars define local rules for agent interactions, facilitating the decentralized creation of desired graph structures, such as $\mathcal{G}_4$. Additionally, while our paper focuses on designing undirected graphs, the approach is adaptable to designing directed networks that are controllable and robust.

This work has applications across various fields, including communications and networked control systems, where balancing network design to induce various properties within the network is crucial. Also, we plan to work on the following extensions in the future.

- Incorporate additional constraints, such as ensuring the existence of a specific subgraph $\mathcal{H}$, which could be a proxy for some other desired network property, while still achieving SSC and maximal robustness in $\mathcal{G}$.
- For controllability, we aim to employ 'energy-based' control metrics as an alternative to solely relying on the binary determination of strong structural controllability. Energy-based metrics offer a more nuanced perspective, quantifying the 'effort' required to control the network, thus providing deeper insights.

Appendix

The appendix presents the partitioning of a given positive integer P into D positive integers: $P_1, P_2, \cdots, P_D$, ensuring $\sum_{i=1}^{D} P_i = P$. We require three inputs: positive integer P, number of partitions D, and a binary input $\mathcal{Z}$ to determine whether partitioning is for the number of leaders or followers. In this context, $\mathcal{Z} = 0$ signifies that P is the number of leaders, and $\mathcal{Z} = 1$ represents that P is the number of followers. This method achieves the required partitioning by modifying the inputs P and D, into $P' = P + 1$ and $D' = D + 1$, respectively, thereby generating D' partitions of P', i.e., $P'_1, P'_2, \cdots, P'_{D'}$. As in optimal clique chains, we consider $P'_1 = P'_{D'} = 1$. Subsequently, $P'_{D'}$ or P'_1 is removed from the final output, depending on whether $\mathcal{Z}$ is zero or one, respectively. This ensures that the ultimate output is D partitions of integer P.

Furthermore, to simplify the partitioning problem, we consider partitioning $\frac{P'}{2}$ into $\frac{D'}{2}$ partitions, such that we get $P'_1, \cdots, P'_{D'/2}$, initially. Then, we replicate this in a mirrored fashion to obtain the other half of the partition, i.e., $\{P'_{(D'/2)+1}, \cdots, P'_{D'}\} = \{P'_{D'/2}, \cdots, P'_1\}$. This guarantees the desired partitioning of P', i.e., $P'_1, \cdots, P'_{D'}$. To proceed, we consider each P'_i, where $i \in \{1, \cdots, n\}$, as a term in the modified geometric series shown below,

$$1 + r + r^2 + \cdots + r^{n-2} + Kr^{n-1} = \left\lceil \frac{P'}{2} \right\rceil, \tag{6}$$

which is,

$$\frac{(1 - r^{n-1})}{(1 - r)} + Kr^{n-1} = \left\lceil \frac{P'}{2} \right\rceil. \tag{7}$$

The terms in geometric series shown in 6 maps to the partition P'_i as follows,

$$P'_i = \begin{cases} r^{i-1} & \text{, if } 1 \leq i \leq n-1 \\ Kr^{i-1} & \text{, if } i = n. \end{cases} \tag{8}$$

Π_1	(r_0)	$a \quad S_{1,1} \quad a \rightharpoonup y_{1,1} \text{——} \ell_{1,1} \text{——} S_{1,2}$	
	(r_1)	$S_{1,i} \quad a \rightharpoonup \ell_{1,i} \text{——} S_{1,i+1}$	$2 \leq i < D$
	(r_2)	$y_{1,i} \quad a \rightharpoonup f_{1,i} \text{——} y_{1,i+1}$	$1 \leq i < D$
	(r_3)	$S_{1,D} \rightharpoonup \ell_{1,D}$	
	(r_4)	$y_{1,D} \rightharpoonup f_{1,D}$	
Π_2	(r_5)	$\ell_{j,i} \quad a \rightharpoonup \ell_{j,i} \text{——} \ell_{j+1,i}$	$\forall\, i,\ 1 \leq j < L_i, \ \text{if } L_i > 1$
	(r_6)	$f_{j,i} \quad a \rightharpoonup f_{j,i} \text{——} f_{j+1,i}$	$\forall\, i,\ 1 \leq j < F_i, \ \text{if } F_i > 1$
	(r_7)	$f_{F_i,i} \quad f_{1,i+1} \rightharpoonup f_{F_i,i} \text{——} f_{1,i+1}$	$1 \leq i < D$
Π_3	(r_8)	$f_{1,i} \text{——} f_{1,i+1} \rightharpoonup f_{1,i} \qquad f_{1,i+1}$	$\begin{cases} i = 1 & \text{if } F_i > 1 \\ 2 \leq i < D & \text{if } F_i > 2 \end{cases}$
Π_4	(r_9)	$f_{F_i-1,i} \quad f_{m,i+1} \rightharpoonup f_{F_i-1,i} \text{——} f_{m,i+1}$	$\forall\, m,\ 2 \leq i < D$
	(r_{10})	$\ell_{j,i} \quad \ell_{m,n} \rightharpoonup \ell_{j,i} \text{——} \ell_{m,n}$	$\forall\, m, \forall\, j, 1 < i < D,\ i-1 \leq\ n\ \leq i+1$
	(r_{11})	$\ell_{j,i} \quad f_{m,n} \rightharpoonup \ell_{j,i} \text{——} f_{m,n}$	$\forall\, m, \forall\, j, 1 < i \leq D, i-2 \leq\ n\ \leq i$

TABLE 2: Graph Grammars for $\mathcal{G}_4$

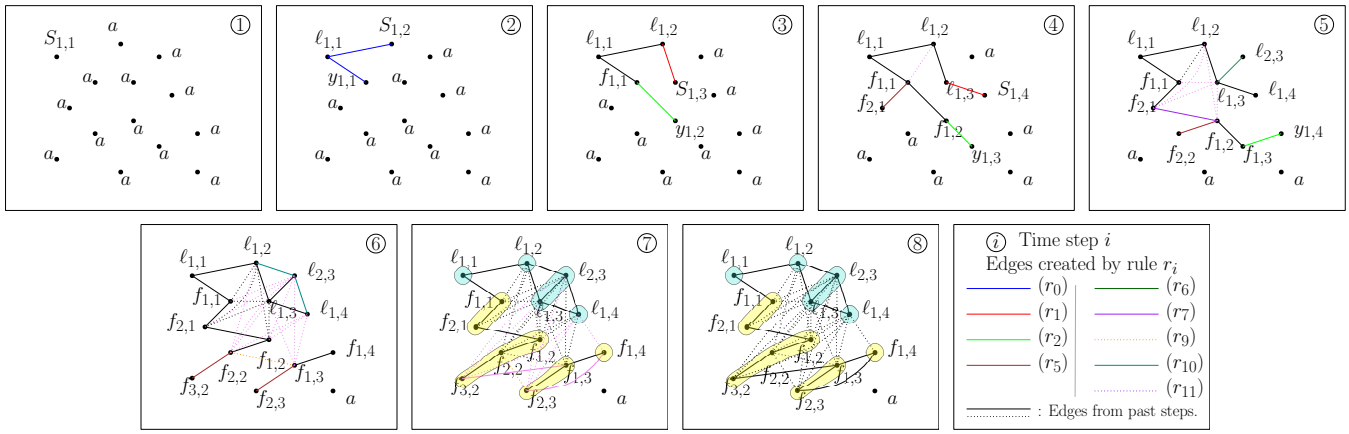


Fig. 14: An illustrative example of constructing $\mathcal{G}_4$ using rules $\mathcal{R}$.

Here, $n = \left\lceil \frac{D'}{2} \right\rceil$. Also, we define the constant K as:

$$K = \begin{cases} \binom{2}{D'-1} & \text{, if } (D'-1) \text{ is even.} \\ \binom{1}{1-\frac{1}{D'-1}} & \text{, if } (D'-1) \text{ is odd.} \end{cases}$$

Now, we can obtain the desired D partitions of integer P from Algorithm 1 and subsequently Algorithm 5, if the following conditions are satisfied after Algorithm 1:

- The sum of partitions $\Sigma_{i=1}^{D'} P'_i$ is equal to the input P'.
- The partitioning $\{P'_1, \cdots, P'_n\}$ is non-decreasing, i.e., $P'_i \leq P'_{i+1}$ and the partitioning $\{P'_{n+1}, \cdots, P'_{D'}\}$ is non-increasing, i.e., $P'_i \geq P'_{i+1}$.
- All the partitions are of at least size one, i.e., $P'_i > 0$.

At the end of Algorithm 1, if the above conditions are not satisfied then we utilize Algorithm 1 in conjunction with Algorithm 2 to obtain the required partitions.

Algorithm 1 Partitioning Algorithm (Part 1)

Input: $P'_i, P, D, \mathcal{Z}$
Output: P_i
1: $P' = P + 1$
2: $D' = D + 1$
3: $n \leftarrow \left\lceil \frac{D'}{2} \right\rceil$
4: **for** $i = 1$ to n **do**
5: $\quad P'_i \leftarrow \text{round}(P'_i)$ ▷ round to the closest integer
6: **end for**
7: $P'_i \leftarrow$ NonDecCheck(P'_i, n)
8: $(P'_i, \text{rem}) \leftarrow$ SymmetricAndSumCheck(P'_i, D', n)

Part 1 of the algorithm plays the most significant role in partitioning the integer P. First, it augments the inputs to P' and D', as mentioned earlier. Then, it rounds the integers P'_i that were obtained after solving Equation 6, ultimately partitioning $\left\lceil \frac{P'}{2} \right\rceil$ integers into $\left\lceil \frac{D'}{2} \right\rceil$ partitions. It is done in such a manner that it follows a non-decreasing distribution. Furthermore, this distribution is then mirrored to the other half of the partitions, ensuring a symmetric distribution around the center partition.

The second segment of the algorithm, presented in Algorithm 2, serves to verify if there is any discrepancy between the input P' and the actual sum of integers P'_i within the partition at that stage of the algorithm. Subsequently, it divides the remaining offset integers in half and makes adjustments to the left half partitions in a manner that maintains their non-decreasing order. Furthermore, this modification is mirrored on the right half partition using Algorithm 4. We also address the scenario where the division of the integer P' into two set of partitions might not be even. Unsystematic subtraction or addition to the partitions may disrupt the desired partition structure. To maintain the integrity of the partitioning arrangement, we carefully implement necessary adjustments.

The partitioning algorithm incorporates three sub-functions to fulfill different objectives. The first sub-function, Algorithm 3, guarantees that the left half partitions $P'_1, \cdots, P'_n$ are of a non-decreasing nature, i.e., $P'_i \leq P'_{i+1}$, provided that the sum of integers to divide is more than the number of partitions i.e., $P' > D'$. This non-decreasing

Algorithm 2 Partitioning Algorithm (Part 2)
(continuation of Partitioning Algorithm (Part 1))

if rem < 0 **then**
  remL $\leftarrow \lfloor \frac{\text{rem}}{2} \rfloor$ remL $\leftarrow \lceil \frac{\text{rem}}{2} \rceil$
end if
$i \leftarrow n$
while remL $\neq 0$ **do**
  $P_U \leftarrow P'_i + \frac{\text{remL}}{|\text{remL}|}$
  if $P_U \geq P'_{i-1}$ **then**
    $P'_i \leftarrow P_U$
    remL $\leftarrow$ remL $- \frac{\text{remL}}{|\text{remL}|}$
  end if
  if $i = 2$ **then**
    $i \leftarrow n$
  else
    $i \leftarrow i - 1$
  end if
end while
$(P'_i, \text{rem}) \leftarrow$ SYMMETRICANDSUMCHECK(P'_i, D', n)
for $i = (n+1)$ to $(D'-1)$ **do**
  $R \leftarrow D' - (i - n)$ ▷ Index for Right half partitions
  if rem < 0 **and** $(P'_i - 1 \geq P'_{i+1}$ **or** $i = D' - 1)$ **then**
    $P'_i \leftarrow P'_i - 1$
  else if rem > 0 **and** $(P'_R + 1 \leq P'_{R-1}$ **or** $R = n + 1)$ **then**
    $P'_R \leftarrow P'_R + 1$
  end if
  rem $\leftarrow P' - \Sigma_{j=1}^{D'} |P'_j|$
end for
$P_i \leftarrow$ AUGMENTEDOUTPUT$(P'_i, \mathcal{Z})$

behavior is essential for subsequent steps in the algorithm.

Algorithm 3 Ensure non-decreasing behaviour in left half

procedure NONDECCHECK(P'_i, n)
  count $\leftarrow 1$
  while count > 0 **do**
    count $\leftarrow 0$
    for $i = n$ to 2 **do**
      if $P'_i < P'_{i-1}$ **and** $P'_{i-1} > 1$ **then**
        $P'_i \leftarrow P'_i + 1$; $P'_{i-1} \leftarrow P'_{i-1} - 1$
        count $\leftarrow$ count $+ 1$
      else if $P'_i < P'_{i-1}$ **and** $P'_{i-1} = 1$ **then**
        $P'_i \leftarrow P'_i + 1$; count $\leftarrow$ count $+ 1$
      else if $P'_{i-1} = 0$ **then**
        $P'_{i-1} \leftarrow 1$
        count $\leftarrow$ count $+ 1$
      end if
    end for
  end while
  return P'_i
end procedure

The second sub-function, Algorithm 4, performs two main tasks. First, it mirrors the partition distribution pattern of the initial half of the partitions, $P'_1, \cdots, P'_n$, to the other half of the partitions, $P'_{n+1}, \cdots, P'_{D'}$, resulting in a symmetric arrangement around the center partition. Secondly, it returns the difference between input P' and the cumulative sum of integers $\Sigma_{i=1}^{D'} P'_i$ allocated up to that point in the algorithm. This information is utilized by the partitioning algorithm for the following operations.

Algorithm 4 Mirror left to right and check total nodes

procedure SYMMETRICANDSUMCHECK(P'_i, D', n)
  $P'_{(D'-(i-1))} \leftarrow P'_i$ ▷ for $i = 1$ to n
  Sum $\leftarrow \Sigma_{i=1}^{D'} P'_i$
  rem $\leftarrow P' -$ Sum
  return (P'_i, rem)
end procedure

The final sub-function Algorithm 5, removes $P'_{D'}$ or P'_1 to obtain the final output P_i, depending on whether $\mathcal{Z}$ pertains to leaders or followers.

Algorithm 5 Removal of Additional Cell

procedure AUGMENTEDOUTPUT$(P'_i, \mathcal{Z})$
  for $i = 1$ to D **do**
    if $\mathcal{Z} = 0$ **then**
      $P_i = P'_i$
    else if $\mathcal{Z} = 1$ **then**
      $P_i = P'_{i+1}$
    end if
  end for
  return P_i
end procedure

REFERENCES

[1] P. Patel, J. Suresh, and W. Abbas, "Distributed design of controllable and robust networks using zero forcing and graph grammars," in *American Control Conference (ACC)*, 2023.

[2] W. Abbas and M. Egerstedt, "Robust graph topologies for networked systems," in *3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys)*, 2012, pp. 85–90.

[3] W. Ellens, F. M. Spieksma, P. Van Mieghem, A. Jamakovic, and R. E. Kooij, "Effective graph resistance," *Linear Algebra and its Applications*, vol. 435, no. 10, pp. 2491–2506, 2011.

[4] D. Zelazo and M. Bürger, "On the robustness of uncertain consensus networks," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 2, pp. 170–178, 2015.

[5] G. F. Young, L. Scardovi, and N. E. Leonard, "A new notion of effective resistance for directed graphs—part I: Definition and properties," *IEEE Transactions on Automatic Control*, vol. 61, pp. 1727–1736, 2016.

[6] H. Wang, R. Kooij, and P. Van Mieghem, "Graphs with given diameter maximizing the algebraic connectivity," *Linear Algebra and its Applications*, vol. 433, no. 11-12, pp. 1889–1908, 2010.

[7] M. Siami and N. Motee, "Fundamental limits on robustness measures in networks of interconnected systems," in *52nd IEEE Conference on Decision and Control*, 2013, pp. 67–72.

[8] W. Abbas, M. Shabbir, A. Y. Yazıcıoğlu, and A. Akber, "Trade-off between controllability and robustness in diffusively coupled networks," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 4, pp. 1891–1902, 2020.

[9] F. Pasqualetti, S. Zhao, C. Favaretto, and S. Zampieri, "Fragility limits performance in complex networks," *Scientific Reports*, vol. 10, no. 1, pp. 1–9, 2020.

[10] W. Abbas, M. Shabbir, Y. Yazıcıoğlu, and X. Koutsoukos, "On zero forcing sets and network controllability–computation and edge augmentation," *IEEE Transactions on Control of Network Systems*, vol. 11, no. 1, pp. 402–413, 2024.

[11] A. Chapman and M. Mesbahi, "On strong structural controllability of networked systems: A constrained matching approach," in *American control conference (ACC)*, 2013, pp. 6126–6131.

[12] M. Shabbir, W. Abbas, Y. Yazicioglu, and X. Koutsoukos, "Computation of the distance-based bound on strong structural controllability in networks," *IEEE Transactions on Automatic Control*, 2022.
[13] J. Jia, H. L. Trentelman, W. Baar, and M. K. Camlibel, "Strong structural controllability of systems on colored graphs," *IEEE Transactions on Automatic Control*, vol. 65, pp. 3977–3990, 2019.
[14] N. Monshizadeh, S. Zhang, and M. K. Camlibel, "Zero forcing sets and controllability of dynamical systems defined on graphs," *IEEE Transactions on Automatic Control*, vol. 59, pp. 2562–2567, 2014.
[15] S. S. Mousavi, M. Haeri, and M. Mesbahi, "On the structural and strong structural controllability of undirected networks," *IEEE Transactions on Automatic Control*, vol. 63, pp. 2234–2241, 2017.
[16] E. Klavins, "Programmable self-assembly," *IEEE Control Systems Magazine*, vol. 27, no. 4, pp. 43–56, 2007.
[17] M. Yim, "Modular self-reconfigurable robot systems: Challenges and opportunities for the future," *IEEE Robotics and Automation Magazine*, vol. 10, pp. 2–11, 2007.
[18] W. Abbas and M. Egerstedt, "Hierarchical assembly of leader-asymmetric, single-leader networks," in *American Control Conference (ACC)*, 2011, pp. 1082–1087.
[19] Y. Yazıcıoğlu, M. Shabbir, W. Abbas, and X. Koutsoukos, "Strong structural controllability of networks: Comparison of bounds using distances and zero forcing," *Automatica*, vol. 146, 2022.
[20] B. Brimkov, C. C. Fast, and I. V. Hicks, "Computational approaches for zero forcing and related problems," *European Journal of Operational Research*, vol. 273, no. 3, pp. 889–903, 2019.
[21] W. Abbas, M. Shabbir, H. Jaleel, and X. Koutsoukos, "Improving network robustness through edge augmentation while preserving strong structural controllability," in *American Control Conference (ACC)*, 2020.
[22] W. Qi, Z. Ji, Y. Liu, and C. Lin, "Strong structural controllability based on leader-follower framework," *Journal of Systems Science and Complexity*, vol. 36, no. 4, pp. 1498–1518, 2023.
[23] X. Chen, S. Pequito, G. J. Pappas, and V. M. Preciado, "Minimal edge addition for network controllability," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 1, pp. 312–323, 2018.
[24] S. S. Mousavi, M. Haeri, and M. Mesbahi, "Strong structural controllability of networks under time-invariant and time-varying topological perturbations," *IEEE Transactions on Automatic Control*, vol. 66, no. 3, pp. 1375–1382, 2020.
[25] ——, "Robust strong structural controllability of networks with respect to edge additions and deletions," in *2017 American Control Conference (ACC)*, 2017, pp. 5007–5012.
[26] C. O. Becker, S. Pequito, G. J. Pappas, and V. M. Preciado, "Network design for controllability metrics," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 3, pp. 1404–1415, 2020.
[27] C. Commault, J. van der Woude, and P. Frasca, "Functional target controllability of networks: Structural properties and efficient algorithms," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 3, pp. 1521–1530, 2019.
[28] G. F. Young, L. Scardovi, and N. E. Leonard, "Robustness of noisy consensus dynamics with directed communication," in *American Control Conference (ACC)*. IEEE, 2010, pp. 6312–6317.
[29] ——, "A new notion of effective resistance for directed graphs—part II: Computing resistances," *IEEE Transactions on Automatic Control*, vol. 61, no. 7, pp. 1737–1752, 2015.
[30] AIM Minimum Rank Special Graphs Work Group, "Zero forcing sets and the minimum rank of graphs," *Linear Algebra and its Applications*, vol. 428, no. 7, pp. 1628–1648, 2008.
[31] H. J. van Waarde, N. Monshizadeh, H. L. Trentelman, and M. K. Camlibel, "Strong structural controllability and zero forcing," *Structural Methods in the Study of Complex Systems*, pp. 91–112, 2020.
[32] P. Barooah and J. P. Hespanha, "Graph effective resistance and distributed control: Spectral properties and applications," in *IEEE Conference on Decision and Control*. IEEE, 2006, pp. 3479–3485.
[33] Y. Yazıcıoğlu, W. Abbas, and M. Shabbir, "Structural robustness to noise in consensus networks: Impact of degrees and distances, fundamental limits, and extremal graphs," *IEEE Transactions on Automatic Control*, vol. 66, no. 10, pp. 4777–4784, 2020.
[34] M. Siami, S. Bolouki, B. Bamieh, and N. Motee, "Centrality measures in linear consensus networks with structured network uncertainties," *IEEE Transactions on Control of Network Systems*, vol. 5, pp. 924–934, 2017.
[35] M. Siami and N. Motee, "Fundamental limits and tradeoffs on disturbance propagation in linear dynamical networks," *IEEE Transactions on Automatic Control*, vol. 61, pp. 4055–4062, 2016.
[36] M. Pirani, E. M. Shahrivar, B. Fidan, and S. Sundaram, "Robustness of leader–follower networked dynamical systems," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 4, 2017.
[37] H. Li and Z. Zhang, "Kirchhoff index as a measure of edge centrality in weighted networks: Nearly linear time algorithms," in *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2018, pp. 2377–2396.
[38] A. Aazami, "Hardness results and approximation algorithms for some problems on graphs," 2008.
[39] B. Brimkov and I. V. Hicks, "Complexity and computation of connected zero forcing," *Discrete Applied Mathematics*, vol. 229, pp. 31–45, 2017.
[40] B. Mckay, "Combinatorial data," http://users.cecs.anu.edu.au/~bdm/data/graphs.html, 2023, accessed: 2023-04-17.
[41] E. Klavins, R. Ghrist, and D. Lipsky, "A grammatical approach to self-organizing robotic systems," *IEEE Transactions on Automatic Control*, vol. 51, no. 6, pp. 949–962, 2006.
[42] A. Zhao, J. Xu, M. Konaković-Luković, J. Hughes, A. Spielberg, D. Rus, and W. Matusik, "Robogrammar: graph grammar for terrain-optimized robot design," *ACM Transactions on Graphics*, vol. 39, pp. 1–16, 2020.
[43] M. Guo, M. Egerstedt, and D. V. Dimarogonas, "Hybrid control of multi-robot systems using embedded graph grammars," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 5242–5247.
[44] B. Smith, A. Howard, J.-M. McNew, J. Wang, and M. Egerstedt, "Multi-robot deployment and coordination with embedded graph grammars," *Autonomous Robots*, vol. 26, pp. 79–98, 2009.

Priyanshkumar Patel is currently working as a Research and Development Engineer at ProPetro Service Inc., Midland, TX, USA. He holds an M.S. degree in Electrical Engineering (2023) from the University of Texas at Dallas, TX, USA. His research interests include multi-agent distributed systems, network robustness, and control.

Johir Suresh is working towards his Ph.D. in automotive engineering at Clemson University, SC, USA. He received M.S. (2023) in Mechanical Engineering from the University of Texas at Dallas, TX, USA, and B.Tech (2020) in automotive engineering from Manipal Academy of Higher Education, Manipal, India. His current research interests include multiagent systems, network robustness, robust and adaptive control, and estimation.

Waseem Abbas is an Assistant Professor in the System Engineering Department at the University of Texas at Dallas, TX, USA. Previously, he was a Research Assistant Professor at the Vanderbilt University, Nashville, TN, USA. He received Ph.D. (2013) and M.Sc. (2010) degrees, both in Electrical and Computer Engineering, from Georgia Institute of Technology, Atlanta, GA, and was a Fulbright scholar from 2009 to 2013. His research interests are in the areas of control of networked systems, resilience and robustness in networks, graph machine learning, and and graph methods in complex networks.