Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 50-1 (2017) 9456–9461

Graph-Based Model Reduction of the Controlled Consensus Protocol ⋆

Noam Leiter * Daniel Zelazo *

* Faculty of Aerospace Engineering, Technion, Haifa 32000 Israel (e-mail: noaml@technion.ac.il, dzelazo@technion.ac.il).

Abstract: We present a general framework for graph-based model reduction of the controlled consensus protocol, and derive a class of edge-based graph contractions as a constructive solution approach. These contractions are utilized in a sub-optimal tree-based greedy-edge reduction method and are demonstrated on the $\mathcal{H}_2$ reduction of the controlled consensus protocol.

Keywords: model reduction, structural properties, multi-agent systems,

1 Introduction

There is currently a rapid growth in the usage of multi-agent systems in many modern technologies. As the complexity and size of such networked systems increases, extensive analysis and simulation become computationally infeasible. Therefore, it is desirable that *reduced order models* are found that still preserve, in some sense, the network and dynamic properties of the original system.

If we consider networked systems as MIMO systems, various reduction methods can be performed at the transfer function level [Benner and Quintana-Ortí (2005)], or directly on the state space realization [Van Dooren et al. (2008)]. In both cases, the resulting reduced order system realizations will generally not be in the functional form of a multi-agent system realization. In [Monshizadeh et al. (2014)], a projection reduction of the controlled consensus protocol is performed based on partitioning of the graph vertices. The resulting reduced order system is then interpreted as an input-output consensus system over a directed graph, and it is shown that the reduction error is $\mathcal{H}_2$-optimal for a special class of partitions. In [Ishizaki et al. (2014)], a similar partition-based projection method is used for reduction of a single input networked dynamic system, resulting in a reduced networked system over a non-simple graph with $\mathcal{H}_\infty$ reduction error bounds.

In this work, we require the model reduction to preserve the functional form of the original controlled consensus system, which results in a reduction of the underlying network structure. As the reduced system preserves a structure of a multi-agent system we can simulate and analyse it with techniques tailored for multi-agent systems.

The main contributions of this work are:

i) The formal definition of graph-based model reduction of the controlled consensus protocol as a graph-based optimization problem.
ii) Derivation of new reduction techniques performed directly on the network structure and of a sub-optimal tree-based greedy-edge reduction algorithm which is computationally efficient.
iii) Demonstration of $\mathcal{H}_2$ graph-based model reduction of the controlled consensus protocol.

We formulate the model reduction over simple weighted graphs as a graph reduction optimization problem. In order to allow a constructive solution, we first restrict the search for optimal graph reductions to the class of graph contractions based on a partition of the graph vertices. Vertex partitions have been extensively studied in graph theory in the context of graph clustering and network communities [Newman and Girvan (2004); Schaeffer (2007); Spielman and Teng (2008)]. The combinatorial nature of such problems requires us to further restrict the graph reduction problem to a class of edge-based contractions and we derive a sub-optimal greedy-edge efficient reduction method.

The remaining sections of this paper are as follows. In Section 2, we formulate the general graph-based reduction of the controlled consensus protocol. In Section 3, graph contractions are presented as a class of graph reductions based on vertex partitions; we introduce the classes of edge-based and tree-based graph contractions and derive two suboptimal efficient graph contraction methods. In Section 4, an analysis of the reduced Laplacian is performed. In Section 5, we construct the $\mathcal{H}_2$ reduction optimization problem of the controlled consensus protocol as a case study for graph-based model reduction. Finally, Section 6 provides some concluding remarks.

Preliminaries A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ consists of a vertex set $\mathcal{V}(\mathcal{G})$, an edge set $\mathcal{E}(\mathcal{G}) = \{\epsilon_1, \dots, \epsilon_{|\mathcal{E}|}\} \subset \mathcal{V}^{[2]}$, and a set of positive edge weights, $\mathcal{W}(\mathcal{G}) = \{w_1, \dots, w_{|\mathcal{E}|}\}$. The order of the graph is defined as the number of nodes. We assign an orientation to the edges using head and tail functions, $h_\mathcal{E}, t_\mathcal{E} : \mathcal{E} \to \mathcal{V}$ where $h_\mathcal{E}(\epsilon_k)$ and $t_\mathcal{E}(\epsilon_k)$ return, respectively, the *head* and *tail* nodes of edge e_k. If $\mathcal{G}$ is an *undirected graph* then the head and tail of each edge are arbitrary; if $\mathcal{G}$ is a *directed graph* (digraph) then the head and tail define the direction of the edge. A *self-loop* is an edge $\epsilon_k \in \mathcal{E}$ such that $h_\mathcal{E}(\epsilon_k) = t_\mathcal{E}(\epsilon_k)$, and *duplicate edges* are any pair $\epsilon_i, \epsilon_j \in \mathcal{E}$ such that $i \neq j$, $t_\mathcal{E}(\epsilon_i) = t_\mathcal{E}(\epsilon_j)$ and $h_\mathcal{E}(\epsilon_i) = h_\mathcal{E}(\epsilon_j)$. A *simple graph* does not include self-loops. A *multi-graph* is a graph that includes duplicate edges. The head and tail functions can be used to define the *incidence function* $f_E : \mathcal{V}(\mathcal{G}) \times \mathcal{E}(\mathcal{G}) \to \{\pm 1, 0\}$, with $f_E(v_i, \epsilon_j) = 1$ if $h_\mathcal{E}(\epsilon_j) = v_i$, $f_E(v_i, \epsilon_j) = -1$ if $t_\mathcal{E}(\epsilon_j) = v_i$, and 0 otherwise. The incidence function can be used to define the corresponding *incidence matrix*, $E(\mathcal{G}) \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{E}|}$, with entries $[E(\mathcal{G})]_{ij} = f_E(v_i, \epsilon_j)$. For a simple undirected graph, the Laplacian matrix $L(\mathcal{G}) \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ is defined as $L(\mathcal{G}) = E(\mathcal{G}) W(\mathcal{G}) E(\mathcal{G})^T$ [Godsil and Royle (2001)]. For a connected graph, the incidence matrix can always be

⋆ The work presented here has been supported by the Israel Science Foundation (grant No. 1490/13).

10.1016/j.ifacol.2017.08.1467

expressed as $E(\mathcal{G}) = [E(\mathcal{T})\ E(\mathcal{C})] = E(\mathcal{T})\underbrace{\left[I_{|\mathcal{T}|}\, T_{(\mathcal{T},\mathcal{C})}\right]}_{=R_{(\mathcal{T},\mathcal{C})}}$, where $T_{(\mathcal{T},\mathcal{C})} = E^L(\mathcal{T})E(\mathcal{C})$,[1] and $\mathcal{T} = (\mathcal{V}, \mathcal{E}_\tau)$ is a spanning tree of $\mathcal{G}$ with $\mathcal{T} \cup \mathcal{C} = \mathcal{G}$ [Zelazo et al. (2013)]. The *essential edge Laplacian* $L_{ess}(\mathcal{G}) \in \mathbb{R}^{n-1\times n-1}$ is the product $L_{ess}(\mathcal{G}) = \hat{L}_e(\mathcal{T})Q(\mathcal{G})$ where $\hat{L}_e(\mathcal{T}) \triangleq E^T(\mathcal{T})E(\mathcal{T})$ and $Q(\mathcal{G}) \triangleq R_{(\mathcal{T},\mathcal{C})}W(\mathcal{G})R^T_{(\mathcal{T},\mathcal{C})}$.

2 Problem Formulation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ be a simple connected undirected graph of order n with a subset $\mathcal{U} \subseteq \mathcal{V}$ of agents subject to external inputs, and let $\mathcal{M} \triangleq (\mathcal{G}, \mathcal{U})$ be the *network structure* . We consider the following *controlled consensus model* over a *network structure* $\mathcal{M}$

$$\Sigma_{\mathcal{M}} \begin{cases} \dot{x} &= -L(\mathcal{G})x + B(\mathcal{U})u \\ y &= W^{\frac{1}{2}}(\mathcal{G})E^T(\mathcal{G})x \end{cases}, \quad (1)$$

where $x \in \mathbb{R}^{|\mathcal{V}|}$ is the system state, $y(t) \in \mathbb{R}^{|\mathcal{E}|}$ are weighted outputs on the edges, $u(t) \in \mathbb{R}^{|\mathcal{U}|}$ are the inputs, and $B(\mathcal{U}) \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{U}|}$ maps each of the inputs to the corresponding leader's node in the network, i.e., $[B(\mathcal{U})]_{ij} = 1$ if v_i is the j'th input node and 0 otherwise.

As shown in [Zelazo et al. (2013)], the system (1) is not a minimal realization; however, applying the transformation $x_\tau = E^T(\mathcal{T}(\mathcal{G}))x$ leads to the so-called *edge agreement protocol* which is minimal. Observe also that the output y includes cycle edges, which was shown to be linearly dependent on the spanning tree. In this direction, for the model reduction we will monitor the signal $z_\tau = Q^{\frac{1}{2}}(\mathcal{G})x_\tau$ corresponding only to the outputs on $\mathcal{T}$ (i.e., we use z_τ in place of the output y in (2) below). The *edge agreement protocol* with the monitor signal z_τ is then

$$\hat{\Sigma}_{\mathcal{M}} \begin{cases} \dot{x}_\tau &= -L_{ess}(\mathcal{G})x_\tau + E^T(\mathcal{T}(\mathcal{G}))B(\mathcal{U})u \\ z_\tau &= Q^{\frac{1}{2}}(\mathcal{G})x_\tau \end{cases}, \quad (2)$$

and the transfer-function matrix (TFM) representation of (2) is,

$$\hat{\Sigma}_{\mathcal{M}}(s) = Q^{\frac{1}{2}}(\mathcal{G})(sI + L_{ess}(\mathcal{G}))^{-1}E^T(\mathcal{T}(\mathcal{G}))B(\mathcal{U}). \quad (3)$$

In this work, we consider the model reduction problem on this minimal realization, where the reduction is performed on the network structure, that is, we approximate a set of n agents with a smaller set of r agents over a reduced graph $\mathcal{G}_r$ with input set $\mathcal{U}_r \subseteq \mathcal{V}_r$. The resulting *reduced network structure*, $\mathcal{M}_r = (\mathcal{G}_r, \mathcal{U}_r)$, is then the underlying structure of the reduced-order edge agreement protocol,

$$\hat{\Sigma}_{\mathcal{M}_r} \begin{cases} \dot{x}_{\tau_r} &= -L_{ess}(\mathcal{G}_r)x_{\tau_r} + E^T(\mathcal{T}(\mathcal{G}_r))B(\mathcal{U}_r)u_r \\ z_{\tau_r} &= Q^{\frac{1}{2}}(\mathcal{G}_r)x_{\tau_r} \end{cases}, \quad (4)$$

and the reduced system TFM, $\hat{\Sigma}_{\mathcal{M}_r}(s)$, is defined analogously to (3). Unlike a standard MIMO model reduction, where the number of inputs and outputs is preserved, the reduced system (4) has a reduced number of outputs and may not preserve the number of inputs.

Let $\mathcal{S}^n_{\mathcal{G}}$ be the set of simple connected graphs, and let $\mathcal{P}_n$ be the power set of n vertices. We define two model reduction optimization problems for some chosen reduction cost function $\mathcal{J}\left(\hat{\Sigma}_{\mathcal{M}_r}, \hat{\Sigma}_{\mathcal{M}}\right)$, which can be formulated in terms of the TFMs or using the state-space realizations.

[1] Here, $E^L = (E(\mathcal{T})^T E(\mathcal{T}))^{-1}E(\mathcal{T})^T$ is the *left-inverse* of $E(\mathcal{T})$.

Problem 1. ***Target structure reduction:***

$$\min_{\mathcal{M}_r \in \left(\mathcal{S}^r_{\mathcal{G}}, \mathcal{P}_r\right)} \mathcal{J}\left(\hat{\Sigma}_{\mathcal{M}_r}, \hat{\Sigma}_{\mathcal{M}}\right). \quad (5)$$

Problem 2. ***Total structure reduction:***

$$\min_{r\le n}\left\{\mathcal{M}_r \in \left(\mathcal{S}^r_{\mathcal{G}}, \mathcal{P}_r\right) | \mathcal{J}\left(\hat{\Sigma}_{\mathcal{M}_r}, \hat{\Sigma}_{\mathcal{M}}\right) \le J_{req}\right\}. \quad (6)$$

The *target structure reduction* requires the optimal reduced model to be of a specified order $r < n$ with minimal reduction error, where the *total structure reduction* specifies a tolerated reduction error and minimizes the reduced model order. In this study, we will focus on solutions to the *target structure reduction*, which may then be used for finding solutions to the *total structure reduction*.

Given a structure $\mathcal{M}$ and an admissible set of reduced graphs $\mathcal{S}^r_{\mathcal{G}}$, finding a solution to Problem 1 may become numerically intractable for a moderate number of nodes, as the number c_r of simple unweighted connected graphs increases exponentially [Wilf (1994)], e.g., for $r = 1, \ldots, 6$, $c_r = 1, 1, 4, 38, 728, 26704$. In the following section we will restrict the class of graph reductions in a way that will allow us to find a suboptimal constructive solution.

3 Reduction by Graph Contractions

The general statement of Problem 1 does not suggest any constructive way to find the optimal structure reduction. However, it is expected that an optimal reduced structure $\mathcal{M}^*_r$ will have some functional dependency on the full structure $\mathcal{M}$. Vertex partitions have been widely used in graph theory, e.g., for graph clustering [Schaeffer (2007)] and in the study of network communities [Newman and Girvan (2004)]. Vertex partitions have been also used for constructing projection-based model reductions of multi-agent systems of the consensus protocol [Monshizadeh et al. (2014)] and bidirectional networks [Ishizaki et al. (2014)]. Here we use vertex partitions as a basis for a constructive method for performing *structure reduction*. We now define several graph operations that will be used in this section.

Definition 1. [Vertex Partition] Let $\mathcal{V}_n = \{1, \ldots, n\}$ and $\mathcal{V}_r = \{1, \ldots, r\}$ be vertex sets of order n and r with $r < n$. We define an *r-partition* $\pi(\mathcal{V}_n) \triangleq \{C_i\}^r_{i=1}$ as the set of r cells with $C_i \subseteq \mathcal{V}_n$, such that $C_i \cap C_j = \emptyset$ and $\cup^r_{i=1} C_i = \mathcal{V}_n$. The corresponding *partition function*, $f_\pi : \mathcal{V}_n \to \mathcal{V}_r$, is defined as $f_\pi(v) \triangleq \{i \in \mathcal{V}_r | v \in C_i\}$ and the corresponding *partition index vector* $p_\pi \in \mathbb{R}^n$ is $[p_\pi]_k = f_\pi(k)$. The set of all *r-partitions* of the vertex set $\mathcal{V}_n$ is $\Pi_r(\mathcal{V}_n) \triangleq \{\{C_i\}^r_{i=1} | C_i \cap C_j = \emptyset,\ \cup^r_{i=1} C_i = \mathcal{V}_n\}$.

Definition 2. [Edge Merging] Let $\mathcal{G} = (\mathcal{V}, \tilde{\mathcal{E}})$ be a multigraph with head and tail functions, $h_{\tilde{\mathcal{E}}}, t_{\tilde{\mathcal{E}}} : \tilde{\mathcal{E}} \to \mathcal{V}$, then the merged edge set $\mathcal{E} = EM(\tilde{\mathcal{E}})$ is the edge set

$$\mathcal{E} = \{\{u, v\} \in \mathcal{V}^2 | \exists\tilde{\epsilon} \in \tilde{\mathcal{E}},\ s.t.\ t_{\tilde{\mathcal{E}}}(\tilde{\epsilon}) = u,\ h_{\tilde{\mathcal{E}}}(\tilde{\epsilon}) = v\}. \quad (7)$$

Definition 3. [Edge Undirecting] Let $\mathcal{G} = (\mathcal{V}, \tilde{\mathcal{E}})$ be a digraph with head and tail functions, $h_{\tilde{\mathcal{E}}}, t_{\tilde{\mathcal{E}}} : \tilde{\mathcal{E}} \to \mathcal{V}$, then the undirected edge set $\mathcal{E} = EUD(\tilde{\mathcal{E}})$ is the edge set

$$\mathcal{E} = \{\{u, v\} \in \mathcal{V}^2 | \exists\tilde{\epsilon} \in \tilde{\mathcal{E}},\ s.t.\ t_{\tilde{\mathcal{E}}}(\tilde{\epsilon}) = u,\ h_{\tilde{\mathcal{E}}}(\tilde{\epsilon}) = v \ \texttt{or} \ h_{\tilde{\mathcal{E}}}(\tilde{\epsilon}) = u,\ t_{\tilde{\mathcal{E}}}(\tilde{\epsilon}) = v\}. \quad (8)$$

Definition 4. [Self-loop Elimination] Let $\mathcal{G} = (\mathcal{V}, \tilde{\mathcal{E}})$ be a graph with head and tail functions, $h_{\tilde{\mathcal{E}}}, t_{\tilde{\mathcal{E}}} : \tilde{\mathcal{E}} \to \mathcal{V}$, then $\mathcal{E} = SLE(\tilde{\mathcal{E}})$ is the edge set obtained by eliminating all

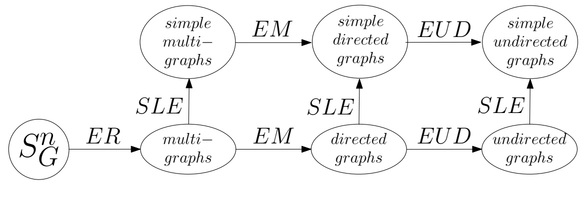


Fig. 1. Graph contraction operator combinations

self-loops in $\tilde{\mathcal{E}}$,

$$\mathcal{E} = \left\{ \tilde{\epsilon} \in \tilde{\mathcal{E}} \,|\, t_{\tilde{\mathcal{E}}}\left(\tilde{\epsilon}\right) \neq h_{\tilde{\mathcal{E}}}\left(\tilde{\epsilon}\right) \right\}. \tag{9}$$

Definition 5. [Edge Reconnecting] Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph of order n with head and tail functions, $h_{\mathcal{E}}, t_{\mathcal{E}} : \mathcal{E} \to \mathcal{V}$, and let $\pi \in \Pi_r(\mathcal{V}(\mathcal{G}))$ with *partition function* f_π. Then the *edge reconnecting* $\tilde{\mathcal{E}}_r = ER(\mathcal{E}, f_\pi)$ is the edge set $\tilde{\mathcal{E}}_r = \{\tilde{\epsilon}_1, \dots, \tilde{\epsilon}_{|\mathcal{E}|}\}$ with head and tail functions, $h_{\tilde{\mathcal{E}}_r}, t_{\tilde{\mathcal{E}}_r} : \tilde{\mathcal{E}}_r \to \mathcal{V}_r$, where $t_{\tilde{\mathcal{E}}_r}(\tilde{\epsilon}_k) = f_\pi(t_{\mathcal{E}}(\epsilon_k))$ and $h_{\tilde{\mathcal{E}}_r}(\tilde{\epsilon}_k) = f_\pi(h_{\mathcal{G}}(\epsilon_k))$ for $\epsilon_k \in \mathcal{E}$.

Given a graph $\mathcal{G} \in \mathcal{S}^n_{\mathcal{G}}$ and a partition $\pi \in \Pi_r(\mathcal{V}(\mathcal{G}))$, we construct the *graph contraction* operation $GC : \mathcal{S}^n_{\mathcal{G}} \to \mathcal{S}^r_{\mathcal{G}}$ as the composition of *edge reconnecting, edge merging, self-loop elimination* and *edge undirecting* operation; this is visualized in Fig.1.

In this study we require the contracted graph to be in the set of simple undirected graphs, and the resulting *graph contraction* operation is given in Algorithm 1. As seen in Fig. 1, it is possible to obtain a simple undirected graph through different sequences of reduction operations. A similar framework can be derived for directed graphs or multi-graphs (both simple and non-simple) by redefining the graph contraction operation.

Algorithm 1 Graph contraction

Input: A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with n vertices and head and tail functions, $h_{\mathcal{G}}, t_{\mathcal{G}} : \mathcal{E} \to \mathcal{V}$, an *r-partition* $\pi \in \Pi_r(\mathcal{V})$

(1) Construct the *partition function* $f_\pi : \mathcal{V} \to \mathcal{V}_r$ and the *partition index vector* $p_\pi \in \mathbb{R}^n$.

(2) Perform *edge reconnecting* and obtain the edge set $\tilde{\mathcal{E}}_r = ER(\mathcal{E}, f_\pi)$.

(3) Perform *edge merging*, followed by *self-loop elimination* and *edge undirecting*, and obtain $\mathcal{E}_r$.

Output: The reduced graph $\mathcal{G}_r = (\mathcal{V}_r, \mathcal{E}_r)$.

Notation: Graph contraction operation - $\mathcal{G}_r = GC(\mathcal{G}, \pi)$.

The following result shows that graph contractions preserve connectedness.

Lemma 6. If $\mathcal{G}$ is connected then the graph contraction $\mathcal{G}_r = GC(\mathcal{G}, \pi_r)$ is connected.

Proof. If $\mathcal{G}$ is connected then $\forall u, v \in \mathcal{V}$, there is a path $uu_1u_2 \dots u_pv$. For any $u^r, v^r \in \mathcal{V}_r$ we can find $u, v \in \mathcal{V}$ such that $f_\pi(u) = u^r$ and $f_\pi(v) = v^r$ and a path $uu_1u_2 \dots u_pv$. If we apply the partition function on the path we obtain a walk (including self loops) in $\mathcal{G}_r$, $u^r f_\pi(u_1) f_\pi(u_2) \dots f_\pi(u_p) v^r$, therefore, $\mathcal{G}_r$ is a connected graph. ■

Using the graph contraction operation we can define corresponding input and structure contractions.

Definition 7. Let $\mathcal{M} = (\mathcal{G}, \mathcal{U})$ and let $\mathcal{G}_r = GC(\mathcal{G}, \pi)$ for $\pi \in \Pi_r(\mathcal{V}(\mathcal{G}))$. The *input contraction* $\mathcal{U}_r = IC(\mathcal{G}_r, \mathcal{U})$ is the set of partition cells containing at least one input node, $\mathcal{U}_r = \{v^r_i \in \mathcal{V}_r \,|\, |\mathcal{U} \cap C_i| > 0, C_i \in \pi\}$, and the resulting *structure contraction* $\mathcal{M}_r = SC(\mathcal{M}, \pi)$ is $\mathcal{M}_r = (\mathcal{G}_r, IC(\mathcal{G}_r, \pi))$.

The graph contractions considered thus far did not include the edge weights. Given a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ and a graph contraction $(\mathcal{V}_r, \mathcal{E}_r) = GC(\mathcal{G}, \pi)$ we are free to choose a new set of weights $\mathcal{W}_r \in \Omega_{W_r} \subseteq \mathbb{R}^{|\mathcal{E}_r|}_+$ in an admissible set Ω_{W_r}. We denote the resulting contracted weighted graph as $\mathcal{G}_r = GC(\mathcal{G}, \pi, \mathcal{W}_r)$, and the corresponding structure contraction as $\mathcal{M}_r = SC(\mathcal{M}, \pi, \mathcal{W}_r)$. Using this notion, the *target structure contraction* problem is to find the optimal *r-partition* and optimal reduced graph edge weights:

Problem 3. ***Target structure contraction***:

$$\min_{\substack{\mathcal{W}_r \in \Omega_{W_r} \\ \pi \in \Pi_r}} \mathcal{J}\left(\hat{\Sigma}_{\mathcal{M}_r}, \hat{\Sigma}_{\mathcal{M}}\right) \tag{10}$$

where $\mathcal{M}_r = SC(\mathcal{M}, \pi, \mathcal{W}_r)$.

We observe that the number of *r-partitions* is $|\Pi_r(\mathcal{G})| = S(n, r)$ where $S(n, r) = \sum_{k=1}^{r} (-1)^{r-k} \frac{k^n}{k!(r-k)!}$ is the Stirling number of the second kind (Wilf, 1994, p.18), which for $r \ll n$ is asymptotically $S(n, r) \sim \frac{r^n}{r!}$.

Definition 8. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with an input set $\mathcal{U} \subseteq \mathcal{V}$, then

i) $\bar{\Pi}_r(\mathcal{V}, \mathcal{U}) \triangleq \{\pi \in \Pi_r(\mathcal{V}) \,|\, |IC(\mathcal{G}_r, \mathcal{U})| = |\mathcal{U}|\}$ is the set of *input invariant r-partitions*.

ii) $\tilde{\Pi}_r(\mathcal{V}, \mathcal{U}) \triangleq \{\pi \in \Pi_r(\mathcal{V}) \,|\, |\mathcal{U} \cap C_j| \leq 1\}$ is the set of *input singleton r-partitions*.

Note that even if we restrict structure contractions to be based on *input singleton r-partitions*, solving Problem 3 is combinatorially hard, and we must further restrict the class of reductions.

3.1 Edge-Based Contractions

The contractions discussed in the previous section are based on vertex partitions. For the reduction of the edge agreement protocol, we might prefer an edge-based reduction method rather than vertex-based, e.g., in [Jongsma et al. (2015)], removal of cycle completing edges was suggested for simplification of the consensus protocol.

Definition 9. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph with *edge contractions* $\mathcal{E}_c = \{e^c_1, \dots, e^c_{n-r}\} \subset \mathcal{E}(\mathcal{G})$ and let $\mathcal{G}_c = (\mathcal{V}, \mathcal{E}_c)$, then we define the *induced edge partition* $\pi_c(\mathcal{G}, \mathcal{E}_c)$ as the set of connected components of $\mathcal{G}_c$. The set of all $n - r$ *edge contractions* is then defined as $\Xi_{n-r}(\mathcal{G}) \triangleq \{\mathcal{E}_c \subset \mathcal{E} \,|\, |\mathcal{E}_c| = n - r\}$.

We propose Algorithm 2 as an edge-based graph contraction algorithm.

In Algorithm 2 the contraction was based on a chosen edge set; however, the actual contraction operation was performed with the induced edge partition. It is possible to derive an equivalent recursive edge contraction algorithm without using the intermediate vertex partition.

Problem 4. ***Target edge-based structure contraction***:

$$\min_{\substack{\mathcal{W}_r \in \Omega_{W_r} \\ \mathcal{E}_c \in \Xi_{n-r}}} \mathcal{J}\left(\hat{\Sigma}_{\mathcal{M}_r}, \hat{\Sigma}_{\mathcal{M}}\right) \tag{11}$$

where $\mathcal{M}_r = SC(\mathcal{M}, \pi_c(\mathcal{G}, \mathcal{E}_c), \mathcal{W}_r)$.

Algorithm 2 Edge-based graph contraction

Input: a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with n vertices, an edge contraction set $\mathcal{E}_c(\mathcal{G}) \in \Xi_{n-r}(\mathcal{G})$.

(1) Construct the graph $\mathcal{G}_c = (\mathcal{V}(\mathcal{G}), \mathcal{E}_c)$.
(2) Calculate the connected components of $\mathcal{G}_c$, and obtain the induced edge partition $\pi_c(\mathcal{G}, \mathcal{E}_c)$.
(3) Perform the graph contraction $\mathcal{G}_r = GC(\mathcal{G}, \pi_c(\mathcal{G}, \mathcal{E}_c))$.

Output: Reduced graph $\mathcal{G}_r$; induced edge partition π_c.
Notation: Edge-based contraction - $(\mathcal{G}_r, \pi_c) = EBC(\mathcal{G}, \mathcal{E}_c)$.

We observe that the number of $n-r$ *edge contractions* is $|\Xi_{n-r}(\mathcal{G})| = \begin{pmatrix} |\mathcal{E}| \\ n-r \end{pmatrix}$.

Definition 10. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with an input set $\mathcal{U} \subseteq \mathcal{V}$, then we define the set of *input free n-r edge contractions* $\hat{\Xi}_{n-r}(\mathcal{G}, \mathcal{U}) \triangleq \{\mathcal{E}_c \in \Xi_{n-r}(\mathcal{G}) \,|\, \mathcal{E}_c \cap \mathcal{U}^2 = \emptyset\}$.

Proposition 11. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with an input set $\mathcal{U} \subseteq \mathcal{V}$, we denote $\mathcal{G}[\mathcal{U}]$ as the input induced subgraph of $\mathcal{G}$ and $d_{\mathcal{U}} = \sum_{k=1}^{m} d(\mathcal{U}_k)$ is the sum of input nodes degrees. Then the number of *input free n-r edge contractions* for $m \leq r$ input nodes is $\left|\hat{\Xi}_{n-r}(\mathcal{G}, \mathcal{U})\right| = \begin{pmatrix} e_{free}(\mathcal{G}, \mathcal{U}) \\ n-r \end{pmatrix}$ where $e_{free}(\mathcal{G}, \mathcal{U}) = |\mathcal{E}(\mathcal{G})| - d_{\mathcal{U}} + |\mathcal{E}(\mathcal{G}[\mathcal{U}])|$.

Proof. The number of edges with input endpoints is the sum of degrees of all input nodes, minus the edges that where counted twice because both endpoints are input nodes, which is the number of edges of the input-induced subgraph of $\mathcal{G}$, therefore, the number of input free edges is $e_{free}(\mathcal{G}, \mathcal{U}) = |\mathcal{E}(\mathcal{G})| - d_{\mathcal{U}} + |\mathcal{E}(\mathcal{G}[\mathcal{U}])|$ and we have $|\hat{\Xi}_{n-r}(\mathcal{G}, \mathcal{U})| = \begin{pmatrix} e_{free}(\mathcal{G}, \mathcal{U}) \\ n-r \end{pmatrix}$. ■

For a complete graph K_n with m input nodes we have $|\mathcal{E}(K_n)| = \frac{n^2-n}{2}$, $d_{\mathcal{U}} = mn$ and $|\mathcal{E}(\mathcal{G}[\mathcal{U}])| = \frac{m^2-m}{2}$, such that $e_{free}(K_n, \mathcal{U}) = \frac{(n-m)^2-(n+m)}{2}$. We conclude that even if we restrict the *edge-based structure contraction* to be based on *input free n-r edge contractions*, solving Problem 4 might still introduce a computational overload, and we must further restrict the class of reductions.

3.2 Tree-Based Contractions

The edges of a spanning tree are the state variables of the edge agreement protocol, therefore, a tree-based reduction method might be advantageous. Here we provide some tree terminology that will assist us in the derivation of a tree-based contraction method.

Definition 12. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a simple connected graph of order n with spanning tree $\mathcal{T}(\mathcal{G})$. Then we denote the set of all spanning trees of $\mathcal{G}$ as $\mathbb{T}(\mathcal{G})$ and the number of spanning trees is defined as $t(\mathcal{G}) \triangleq |\mathbb{T}(\mathcal{G})|$.

Proposition 13. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a simple connected graph of order n and let $\mathcal{T} \in \mathbb{T}(\mathcal{G})$ with an $n-r$ edge contraction set $\mathcal{E}_c(\mathcal{T}) \in \Xi_{n-r}(\mathcal{T})$, and let $(\mathcal{G}_r, \pi_c) = EBC(\mathcal{T}, \mathcal{E}_c)$. Then $\mathcal{G}_r$ is a tree $\mathcal{T}_r$ of order r.

Proof. A tree of order n has $n-1$ edges, and by contracting $n-r$ tree edges we are left with $(n-1)-(n-r)$ edges, such that $|\mathcal{E}(\mathcal{G}_r)| = r-1$. From Lemma 6 we obtain that $\mathcal{G}_r$ is connected, therefore, $\mathcal{G}_r$ is a connected graph of order r with $r-1$ edges, which is a tree of order r. ■

Since $\mathcal{G}$ and $\mathcal{T} \in \mathbb{T}(\mathcal{G})$ share the same vertices, they have equal partitions, $\Pi_r(\mathcal{G}) = \Pi_r(\mathcal{T})$. Given a graph $\mathcal{G}$, if we perform an edge-based contraction of $\mathcal{T}(\mathcal{G})$, $(\mathcal{T}_r, \pi_c) = EBC(\mathcal{T}, \mathcal{E}_c)$, we obtain π_c as a partition of $\mathcal{G}$. This leads us to a tree-based graph contraction operation given in Algorithm 3.

Algorithm 3 Tree-based graph contraction

Input: a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of order n, a tree $\mathcal{T} \in \mathbb{T}(\mathcal{G})$, and an $n-r$ edge contraction set $\mathcal{E}_c(\mathcal{T}) \in \Xi_{n-r}(\mathcal{T})$

(1) Perform the edge-based graph contraction $(\mathcal{T}_r, \pi_c) = EBC(\mathcal{T}, \mathcal{E}_c(\mathcal{T}))$.
(2) Perform the graph contraction of $\mathcal{G}$ with π_c, $\mathcal{G}_r = GC(\mathcal{G}, \pi_c)$.

Output: Reduced graph $\mathcal{G}_r$; tree $\mathcal{T}_r$; partition π_c
Notation: Tree-based contraction - $(\mathcal{G}_r, \mathcal{T}_r, \pi_c) = TBC(\mathcal{G}, \mathcal{T}, \mathcal{E}_c(\mathcal{T}))$.

We notice that the outputs $\mathcal{T}_r$ and $\mathcal{G}_r$ of Algorithm 3 share the same vertices, therefore, $\mathcal{T}_r$ is a spanning tree of $\mathcal{G}_r$, $\mathcal{T}(\mathcal{G}_r) = \mathcal{T}_r$. The input-free tree-based contraction optimization problem is then to find the optimal spanning tree $\mathcal{T}^* \in \mathbb{T}(\mathcal{G})$ and the optimal $n-r$ input-free edge contraction set $\mathcal{E}_c^* \in \hat{\Xi}_{n-r}(\mathcal{T}^*)$ that minimize the reduction cost.

Problem 5. ***Target tree-based structure contraction***:

$$\min_{\substack{\mathcal{W}_r \in \Omega_{W_r} \\ \mathcal{E}_c \in \hat{\Xi}_{n-r}(\mathcal{T}, \mathcal{U}) \\ \mathcal{T} \in \mathbb{T}(\mathcal{G})}} \mathcal{J}\left(\hat{\Sigma}_{\mathcal{M}_r}, \hat{\Sigma}_{\mathcal{M}}\right),$$

where $\mathcal{M}_r = SC(\mathcal{M}, \pi_c(\mathcal{T}, \mathcal{E}_c), \mathcal{W}_r)$.

From Proposition 11 we obtain that $\left|\hat{\Xi}_{n-r}(\mathcal{T}, \mathcal{U})\right| = \begin{pmatrix} e_{free}(\mathcal{T}, \mathcal{U}) \\ n-r \end{pmatrix}$ with $e_{free}(\mathcal{T}, \mathcal{U}) = n-1-d_{\mathcal{U}}+|\mathcal{E}(\mathcal{T}[\mathcal{U}])|$.

The number of contractions is then $t(\mathcal{G}) \times \begin{pmatrix} e_{free}(\mathcal{T}, \mathcal{U}) \\ n-r \end{pmatrix}$ and Problem 5 might still be unfeasible. Here we suggest the greedy tree-based contraction algorithm as a sub-optimal contraction method.

In Algorithm 4, we do not specify the choice of spanning tree and the algorithm can be repeated to other spanning trees in the graph.

4 The Reduced Laplacian Matrix

The Laplacian matrix plays a key role in the controlled consensus protocol. In this section we present the general algebraic relation between the Laplacian and essential edge Laplacian matrices of the full graph $\mathcal{G}$ and those of the reduced graph $\mathcal{G}_r = GC(\mathcal{G}, \pi)$.

Definition 14. [partition characteristic matrix] Let π be an *r-partition* of a vertex set $\mathcal{V}$ with partition function f_π, we define $P_\pi \in \mathbb{R}^{n \times r}$, the *partition characteristic matrix* with $[P_\pi]_{ij} = 1$ if $f_\pi(v_i) = v_j^r$, and 0 otherwise.

Under the graph contraction operation, each edge of the full graph that connects two vertices in two different partition cells is mapped to an edge in the reduced graph, and each edge connecting two vertices in the same cell is mapped to a self loop and omitted. We now define the edge reduction matrix.

Definition 15. [Edge reduction matrix] Let $\mathcal{G}_r = GC(\mathcal{G}, \pi)$ with arbitrary head and tail functions $h_{\mathcal{E}_r}, t_{\mathcal{G}_r}$. Then the *edge reduction matrix*, $P_{(\mathcal{E}, \mathcal{E}_r)} \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{E}_r|}$, is defined

Algorithm 4 Input-free greedy tree-based contraction

Input: a structure $\mathcal{M} = (\mathcal{G}, \mathcal{U})$ with $\mathcal{G}$ of order n, a tree $\mathcal{T} \in \mathbb{T}(\mathcal{G})$, reduction cost function $\mathcal{J}(\hat{\Sigma}_{\mathcal{M}_r}, \hat{\Sigma}_{\mathcal{M}})$, required reduction order r

(1) For $k = 1, 2, \ldots, n-1$ **do:**
   (a) check if $e_k^{\mathcal{T}} \in \mathcal{T}$ includes an input node, if yes, set $\mathcal{J}_k = \infty$ and skip to the next edge.
   (b) perform the single tree-based graph contraction $\left(\mathcal{G}_{n-1}^k, \mathcal{T}_{n-1}^k, \pi_c^k\right) = TBC\left(\mathcal{G}, \mathcal{T}, e_k^{\mathcal{T}}\right)$.
   (c) obtain the structure contraction $\mathcal{M}_{n-1}^k = SC\left(\mathcal{M}, \pi_c^k, \mathcal{W}\left(\mathcal{G}_{n-1}^k\right)\right)$.
   (d) calculate the reduction cost $\mathcal{J}_k^* = \min_{\mathcal{W}(\mathcal{G}_{n-1}^k)} \mathcal{J}\left(G_{\mathcal{M}_{n-1}^k}, G_{\mathcal{M}}\right)$.

(2) Construct the contraction set $\mathcal{E}_c(\mathcal{T})$ from the $n-r$ edges with lowest reduction cost in $\{\mathcal{J}_k^*\}_{k=1}^{n-1}$.

(3) Perform the tree-based contraction $(\mathcal{G}_r, \mathcal{T}_r, \pi_c) = TBC(\mathcal{G}, \mathcal{T}, \mathcal{E}_c(\mathcal{T}))$.

(4) Obtain the structure contraction $\mathcal{M}_r = SC(\mathcal{M}, \pi_c, \mathcal{W}(\mathcal{G}_r))$.

(5) Calculate the reduction cost $\mathcal{J}_{GTC} = \min_{\mathcal{W}(\mathcal{G}_r)} \mathcal{J}(G_{\mathcal{M}_r}, G_{\mathcal{M}})$.

Output: Graph $\mathcal{G}_r$; tree $\mathcal{T}_r$; partition π_c; reduction cost $\mathcal{J}_{GTC}$

Notation: We denote the greedy tree contraction as $(\mathcal{G}_r, \mathcal{T}_r, \pi_c, \mathcal{J}_{GTC}) = GTC(\mathcal{M}, \mathcal{T}, \mathcal{J}(,), r)$.

such that $\left[P_{(\mathcal{E},\mathcal{E}_r)}\right]_{km} = 1$ if $f_\pi(h_\mathcal{E}(\epsilon_k)) = h_{\mathcal{E}_r}(\epsilon_m^r)$, $f_\pi(t_\mathcal{E}(\epsilon_k)) = t_{\mathcal{E}_r}(\epsilon_m^r)$, $\left[P_{(\mathcal{E},\mathcal{E}_r)}\right]_{km} = -1$ if $f_\pi(h_\mathcal{E}(\epsilon_k)) = t_{\mathcal{E}_r}(\epsilon_m^r)$, $f_\pi(t_\mathcal{E}(\epsilon_k)) = h_{\mathcal{E}_r}(\epsilon_m^r)$, and 0 otherwise.

The corresponding *normalized edge reduction matrix* is $U_{(\mathcal{E},\mathcal{E}_r)} \triangleq P_{(\mathcal{E},\mathcal{E}_r)} D_{\mathcal{E}_r}^{-1}$ with $D_{\mathcal{E}_r} \triangleq P_{(\mathcal{E},\mathcal{E}_r)}^T P_{(\mathcal{E},\mathcal{E}_r)}$, a diagonal matrix where each entry on the diagonal is the number of edges mapped to an edge in the reduced graph.

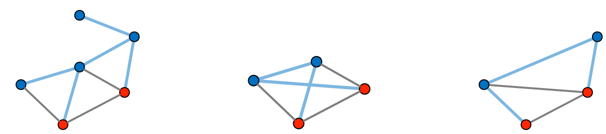
Proposition 16. Let $\mathcal{G}$ be an n order simple graph and let $\mathcal{G}_r = GC(\mathcal{G}, \pi, \mathcal{W}_r)$ be a graph contraction of $\mathcal{G}$ for $\pi \in \Pi_r(\mathcal{G})$ and chosen edge weights $\mathcal{W}_r$. Then the graph contraction incidence matrix for arbitrary head and tail functions $h_{\mathcal{E}_r}, t_{\mathcal{E}_r}$ takes the form

$$E(\mathcal{G}_r) = P_\pi^T E(\mathcal{G}) U_{(\mathcal{E},\mathcal{E}_r)}, \tag{12}$$

and the reduced Laplacian matrix is

$$L(\mathcal{G}_r) = P_\pi^T E(\mathcal{G}) U_{(\mathcal{E},\mathcal{E}_r)} W(\mathcal{G}_r) U_{(\mathcal{E},\mathcal{E}_r)}^T E^T(\mathcal{G}) P_\pi. \tag{13}$$

Proof. Let $e_j \in \mathbb{R}^n$ be the j'th column of $E(\mathcal{G})$ corresponding to edge $\epsilon_j \in \mathcal{E}(\mathcal{G})$, $[e_j]_k = 1$ if $h_\mathcal{E}(\epsilon_j) = v_k$, $[e_j]_k = -1$ if $t_\mathcal{E}(\epsilon_j) = v_k$ and 0 otherwise, and let $p_i \in \mathbb{R}^n$ be the i'th column of P_π corresponding to the i'th cell $C_i \in \pi$, $[p_i]_k = 1$ if $[p_i]_k$ and 0 otherwise. Then we obtain that $p_i^T e_j = 1$ if $t_\mathcal{E}(\epsilon_j) \notin C_i$, $h_\mathcal{E}(\epsilon_j) \in C_i$, $p_i^T e_j = -1$ if $t_\mathcal{E}(\epsilon_j) \in C_i$, $h_\mathcal{E}(\epsilon_j) \notin C_i$, and $p_i^T e_j = 0$ otherwise. Let $\tilde{\mathcal{G}}_r = \left(\tilde{\mathcal{E}}_r, \mathcal{V}_r\right)$ be the r order multi-graph with $\tilde{\mathcal{E}}_r = ER(\mathcal{E}, f_\pi)$ the *edge reconnecting* (Definition 5), where f_π is the partition function of π. From the *edge reconnecting* definition of head and tail, $t_{\tilde{\mathcal{E}}_r}(\tilde{\epsilon}_k) = f_\pi(t_\mathcal{E}(\epsilon_k))$ and $h_{\tilde{\mathcal{E}}_r}(\tilde{\epsilon}_k) = f_\pi(h_\mathcal{G}(\epsilon_k))$ for $\epsilon_k \in \mathcal{E}(\mathcal{G})$, we can construct $\tilde{e}_j \in \mathbb{R}^r$, the j'th column of $E\left(\tilde{\mathcal{G}}\right)$, such



(a) Graph with 6 nodes, 8 edges, and 2 inputs (red). (b) Optimal contraction ($\mathcal{J}_2 = 0.0711$). (c) Optimal edge-based contraction ($\mathcal{J}_2 = 0.2266$).

Fig. 2. $\mathcal{H}_2$ reduction of a controlled consensus protocol.

that $[\tilde{e}_j]_k = 1$ if $t_{\tilde{\mathcal{E}}}(\tilde{\epsilon}_j) \neq v_k^r$, $h_{\tilde{\mathcal{E}}}(\tilde{\epsilon}_j) = v_k^r$, $[\tilde{e}_j]_k = -1$ if $t_{\mathcal{E}}(\tilde{\epsilon}_j) = v_k^r$, $h_{\tilde{\mathcal{E}}}(\tilde{\epsilon}_j) \neq v_k^r$, and $[\tilde{e}_j]_k = 0$ otherwise.

We obtain that $[\tilde{e}_j]_k = p_i^T e_j$, and therefore, $E(\tilde{\mathcal{G}}_r) = P_\pi^T E(\mathcal{G})$ where P_π is the partition characteristic matrix. The reduced graph $\mathcal{G}_r = GC(\mathcal{G}, \pi)$ is obtained from $\tilde{\mathcal{G}}_r = (\tilde{\mathcal{E}}_r, \mathcal{V}_r)$ by performing *edge merging*, *self-loop elimination* and *edge undirecting*. It is readily seen from the definition of the *edge reduction matrix* that $U_{(\mathcal{E},\mathcal{E}_r)}$ operates on the columns of $E(\tilde{\mathcal{G}}_r)$ the algebraic equivalence of *edge merging*, *self-loop elimination* and *edge undirecting*, such that $E(\mathcal{G}_r) = E(\tilde{\mathcal{G}}_r) U_{(\mathcal{E},\mathcal{E}_r)} = P_\pi^T E(\mathcal{G}) U_{(\mathcal{E},\mathcal{E}_r)}$. With $E(\mathcal{G}_r) = P_\pi^T E(\mathcal{G}) U_{(\mathcal{E},\mathcal{E}_r)}$, $L(\mathcal{G}_r) = E(\mathcal{G}_r) W(\mathcal{G}_r) E^T(\mathcal{G}_r)$ takes the form of (13). ■

Without loss of generality, we can order the edges of the graph and the reduced graph to spanning tree edges and cycle completing edges $\mathcal{E}(\mathcal{G}) = \mathcal{E}(\mathcal{T}) \cup \mathcal{E}(\mathcal{C})$ such that the *edge reduction matrix* takes the form

$$P_{(\mathcal{E},\mathcal{E}_r)} = \begin{bmatrix} P_{(\mathcal{T},\mathcal{T}_r)} & P_{(\mathcal{T},\mathcal{C}_r)} \\ P_{(\mathcal{C},\mathcal{T}_r)} & P_{(\mathcal{C},\mathcal{C}_r)} \end{bmatrix}, \tag{14}$$

where $P_{(\mathcal{T},\mathcal{T}_r)} \in \mathbb{R}^{|\mathcal{E}(\mathcal{T})| \times |\mathcal{E}(\mathcal{T}_r)|}$, $P_{(\mathcal{T},\mathcal{C}_r)} \in \mathbb{R}^{|\mathcal{E}(\mathcal{T})| \times |\mathcal{E}(\mathcal{C}_r)|}$, $P_{(\mathcal{C},\mathcal{C}_r)} \in \mathbb{R}^{|\mathcal{E}(\mathcal{C})| \times |\mathcal{E}(\mathcal{C}_r)|}$ and $P_{(\mathcal{C},\mathcal{T}_r)} \in \mathbb{R}^{|\mathcal{E}(\mathcal{C})| \times |\mathcal{E}(\mathcal{T}_r)|}$ are the corresponding tree edges and cycle completing edges reduction matrices. We define the two diagonal matrices $D_{\mathcal{T}_r} \triangleq P_{(\mathcal{T},\mathcal{T}_r)}^T P_{(\mathcal{T},\mathcal{T}_r)} + P_{(\mathcal{C},\mathcal{T}_r)}^T P_{(\mathcal{C},\mathcal{T}_r)}$ and $D_{\mathcal{C}_r} \triangleq P_{(\mathcal{T},\mathcal{C}_r)}^T P_{(\mathcal{T},\mathcal{C}_r)} + P_{(\mathcal{C},\mathcal{C}_r)}^T P_{(\mathcal{C},\mathcal{C}_r)}$, such that $D_{\mathcal{E}_r} = \mathrm{diag}(D_{\mathcal{T}_r}, D_{\mathcal{C}_r})$ and the *normalized edge reduction matrix* takes the form

$$U_{(\mathcal{E},\mathcal{E}_r)} = \begin{bmatrix} P_{(\mathcal{T},\mathcal{T}_r)} D_{\mathcal{T}_r}^{-1} & P_{(\mathcal{T},\mathcal{C}_r)} D_{\mathcal{C}_r}^{-1} \\ P_{(\mathcal{C},\mathcal{T}_r)} D_{\mathcal{T}_r}^{-1} & P_{(\mathcal{C},\mathcal{C}_r)} D_{\mathcal{C}_r}^{-1} \end{bmatrix}. \tag{15}$$

Using $E(\mathcal{G}_r) = P_\pi^T E(\mathcal{G}) U_{(\mathcal{E},\mathcal{E}_r)}$ and $E(\mathcal{G}_r) = [E(\mathcal{T}_r)\ E(\mathcal{C}_r)]$ we get $E(\mathcal{G}_r) = [E(\mathcal{T}_r)\ E(\mathcal{C}_r)] = P_\pi^T [E(\mathcal{T})\ E(\mathcal{C})] U_{(\mathcal{E},\mathcal{E}_r)}$, from which we obtain $E(\mathcal{T}_r) = P_\pi^T E(\mathcal{T}) T_{(\mathcal{T},\mathcal{T}_r)}$ and $E(\mathcal{C}_r) = P_\pi^T E(\mathcal{T}) T_{(\mathcal{T},\mathcal{C}_r)}$, where we define

$$T_{(\mathcal{T},\mathcal{T}_r)} \triangleq \left(P_{(\mathcal{T},\mathcal{T}_r)} + T_{(\mathcal{T},\mathcal{C})} P_{(\mathcal{C},\mathcal{T}_r)}\right) D_{\mathcal{T}_r}^{-1}, \tag{16}$$

$$T_{(\mathcal{T},\mathcal{C}_r)} \triangleq \left(P_{(\mathcal{T},\mathcal{C}_r)} + T_{(\mathcal{T},\mathcal{C})} P_{(\mathcal{C},\mathcal{C}_r)}\right) D_{\mathcal{C}_r}^{-1}. \tag{17}$$

The reduced essential edge Laplacian is then $L_{ess}(\mathcal{G}_r) = T_{(\mathcal{T},\mathcal{T}_r)}^T E^T(\mathcal{T}) P_\pi P_\pi^T E(\mathcal{T}) T_{(\mathcal{T},\mathcal{T}_r)} Q(\mathcal{G}_r)$.

5 $\mathcal{H}_2$ Graph-Based Model Reduction of the Controlled Consensus Protocol

In this section, we study a reduction cost function based on the $\mathcal{H}_2$ performance of the controlled consensus protocol. In order to examine the reduction error we compare the outputs of the full and reduced systems. Using the tree to reduced tree mapping $T_{(\mathcal{T},\mathcal{T}_r)}$ (16) we can define a difference signal $x_\tau - T_{(\mathcal{T},\mathcal{T}_r)}x_{\tau_r}$, and an output difference signal $Q^{\frac{1}{2}}(\mathcal{G})\left(x_\tau - T_{(\mathcal{T},\mathcal{T}_r)}x_{\tau_r}\right)$. We can then construct a reduction error system with augmented state $x_e = \left[\, x_\tau^T \; x_{\tau_r}^T \,\right] \in \mathbb{R}^{n+r-2}$,

$$\Sigma_e(\mathcal{M},\mathcal{M}_r) \begin{cases} \dot{x}_e &= A_e x_e + B_e u \\ z_e &= C_e x_e \end{cases}, \qquad (18)$$

where $A_e = \text{Diag}\{-L_{ess}(\mathcal{G}), -L_{ess}(\mathcal{G}_r)\}$,

$B_e = \left[B^T(\mathcal{U})\, E(\mathcal{T})\, B^T(\mathcal{U}_r)\, E(\mathcal{T}_r)\right]^T$, $C_e = \left[Q^{\frac{1}{2}}(\mathcal{G}) - Q^{\frac{1}{2}}(\mathcal{G})\, T_{(\mathcal{T},\mathcal{T}_r)}\right]$

Performing a reduction based on a contraction $\pi \in \bar{\Pi}_r(\mathcal{G},\mathcal{U})$, we define a reduction error cost using the $\mathcal{H}_2$-norm of system (18),

$$\mathcal{J}_2(G_{\mathcal{M}_r}, G_{\mathcal{M}}) = \frac{\|\Sigma_e(\mathcal{M},\mathcal{M}_r)\|^2_{\mathcal{H}_2}}{\|\hat{\Sigma}(\mathcal{M})\|^2_{\mathcal{H}_2}}. \qquad (19)$$

Given a stable linear system with state-space realization $\Sigma : (A, B, C)$, the $\mathcal{H}_2$-norm of Σ can be calculated from $\|\Sigma\|_{\mathcal{H}_2} = \sqrt{\text{Tr}(B^T X B)}$ with X the solution to the Lyapunov equation $A^T X + XA + C^T C = 0$.

Proposition 17. Let $\mathcal{M} = (\mathcal{G},\mathcal{U})$ with $\mathcal{G}$ of order n and $|\mathcal{U}| = m$. Then the $\mathcal{H}_2$ performance of the *edge agreement protocol* (2) is $\|\hat{\Sigma}_{\mathcal{M}}\|^2_{\mathcal{H}_2} = \frac{m}{2}\left(1 - \frac{1}{n}\right)$.

Proof. With $A = -L_{ess}(\mathcal{G})$ and $C = Q^{\frac{1}{2}}(\mathcal{G})$ the solution to the Lyapunov equation is $X = \frac{1}{2}\hat{L}_e^{-1}$. Substituting X and $E^T(\mathcal{T}(\mathcal{G}))\,B(\mathcal{U})$ in $\text{Tr}\left(B^T X B\right)$ we get $\left\|\hat{\Sigma}(\mathcal{M})\right\|^2_{\mathcal{H}_2} = \frac{1}{2}\text{Tr}\left(B(\mathcal{U})^T E(\mathcal{T})\left[\hat{L}_e(\mathcal{T})\right]^{-1} E^T(\mathcal{T})\,B(\mathcal{U})\right)$. From $\hat{L}_e = E^T(\mathcal{T})\,E(\mathcal{T})$ and using $E(\mathcal{T})\,\hat{L}_e^{-1}E^T(\mathcal{T}) = I_{n\times n} - \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T$, with $\mathbb{1}_n \in \mathbb{R}^n$ a vector of ones, we obtain $\|\hat{\Sigma}(\mathcal{M})\|^2_{\mathcal{H}_2} = \frac{1}{2}\text{Tr}\left(B(\mathcal{U})^T B(\mathcal{U})\right) - \frac{\left\|B(\mathcal{U})^T\mathbb{1}_n\right\|^2}{2n}$. From the definition of $B(\mathcal{U})$ we obtain $\|\hat{\Sigma}(\mathcal{M})\|^2_{\mathcal{H}_2} = \frac{m}{2}\left(1 - \frac{1}{n}\right)$. ■

Similarly to the full system, the $\mathcal{H}_2$ performance of the reduced system (4) is $\|\hat{\Sigma}(\mathcal{M}_r)\|^2_{\mathcal{H}_2} = \frac{m}{2}\left(1 - \frac{1}{r}\right)$.

For the reduction error system (18) we obtain $\|\Sigma_e\|^2_{\mathcal{H}_2} = \text{Tr}\left(B_e^T X B_e\right)$ with X the solution to the Lyapunov equation $A_e^T X + XA_e + C_e^T C_e = 0$.

5.1 Case Studies

As a first example, consider the controlled consensus system (2) over a graph of order 6 with 8 unit-weight edges and two input nodes (Fig.2a).

We require the reduced system to be of order 4 with unit edge weights. In this case, the number of possible partitions is $S(6,4) = 65$ and we can find the optimal contraction (Fig.(2b)). There are $\binom{6}{2} = 15$ possible edge-based contractions, and we can obtain the optimal one (Fig.(2c)). Choosing a spanning tree (solid blue edges), we perform the greedy input-free tree-based contraction (Algorithm (4)), and obtain the exact same reduced graph as the optimal edge-based contraction in Fig. (2c).

Next we apply the greedy tree-based algorithm 4 on the consensus system over a Buckminster Fuller ("Bucky") graph with 5 inputs (Fig.3a). The Bucky graph is of order 60 with 90 unit-weight edges and we require the reduced

(a) Bucky graph with 5 inputs. (b) Reduced graph of order 30 with 49 edges.

Fig. 3. Reduction of the consensus system over a Bucky graph.

graph to be of order 30 with unit-weight edges. In order to find the optimal structure contraction for this case we would have to examine $S(60,30) = 9.5635 \cdot 10^{53}$ possible contractions. If we restrict to edge-based contractions we have $\binom{90}{30} = 6.7313 \cdot 10^{23}$ cases. Therefore, a sub-optimal reduced structure is obtained with the greedy tree-based contraction (Algorithm 4), and we obtain the reduction error $\mathcal{J}_2 = 0.0294$ (Fig.3b).

6 Conclusions

We have defined the graph-based model reduction of the controlled-consensus protocol. The greedy tree-based contraction algorithm has been suggested as a suboptimal efficient solution and demonstrated with an $\mathcal{H}_2$ reduction error.

The same graph-based model reduction framework can be applied to the reduction of other multi-agents system. Various other suboptimal structure reduction methods can be derived for the problem, as well as methods for optimization of non-unit edge-weights. Other reduction error metrics may be considered for the reduced consensus system.

References

Benner, P. and Quintana-Ortí, E.S. (2005). *Model reduction based on spectral projection methods*. Springer.

Godsil, C. and Royle, G. (2001). *Algebraic Graph Theory*. Springer.

Ishizaki, T., Kashima, K., Imura, J.i., and Aihara, K. (2014). Model reduction and clusterization of large-scale bidirectional networks. *IEEE Transactions on Automatic Control*, 59(1), 48–63.

Jongsma, H., Trentelman, H., and Camlibel, M. (2015). Model reduction of consensus networks by graph simplification. In *2015 54th IEEE Conference on Decision and Control (CDC)*, 5340–5345. IEEE.

Monshizadeh, N., Trentelman, H.L., and Camlibel, M.K. (2014). Projection-based model reduction of multi-agent systems using graph partitions. *IEEE Transactions on Control of Network Systems*, 1(2), 145–154.

Newman, M.E. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical review E*, 69(2), 026113.

Schaeffer, S.E. (2007). Graph clustering. *Computer science review*, 1(1), 27–64.

Spielman, D.A. and Teng, S.H. (2008). A local clustering algorithm for massive graphs and its application to nearly-linear time graph partitioning. *arXiv preprint arXiv:0809.3232*.

Van Dooren, P., Gallivan, K.A., and Absil, P.A. (2008). $\mathcal{H}_2$-optimal model reduction of mimo systems. *Applied Mathematics Letters*, 21(12), 1267–1273.

Wilf, H. (1994). *Generatingfunctionology*. Academic Press Inc - New York.

Zelazo, D., Schuler, S., and Allgöwer, F. (2013). Performance and design of cycles in consensus networks. *Systems & Control Letters*, 62(1), 85–96.