Efficient Model Order Reduction for Multi-Agent Systems Using QR Decomposition-Based Clustering

Petar Mlinarić[1,2] Sara Grundel[1] Peter Benner[1]

***Abstract*— In this paper we present an efficient model order reduction method for multi-agent systems with Laplacian-based dynamics. The method combines an established model order reduction method and a clustering algorithm to produce a graph partition used for reduction, thus preserving structure and consensus. By the Iterative Rational Krylov Algorithm, a good reduced order model can be found which is not necessarily structure preserving. However, based on this we can efficiently find a partition using the QR decomposition with column pivoting as a clustering algorithm, so that the structure can be restored. We illustrate the effectiveness on an example from the open literature.**

I. Introduction

Simulation and control of systems connected in a network are ubiquitous problems in many areas of the sciences and engineering, see, e.g., [1], [2], [3]. Due to their complexity, networked systems usually constitute large-scale dynamical systems. Model order reduction methods are often necessary for analysis, simulation, and control of large-scale systems. Additionally, for multi-agent systems, model order reduction methods preserving network structure and consensus (see [4] for an overview of the consensus problem) are preferred.

References [5], [6], [7] introduce *leader-invariant equitable partitions* and corresponding *quotient graphs*, which can be used for eliminating some uncontrollable states, thus performing no model reduction error. Reference [8] proposes *leader partitions*, which introduce "only small model errors". The authors of [9] develop a clustering-based $\mathcal{H}_\infty$ model order reduction method based on *positive tridiagonalization* and *θ-reducible clusters*, applicable to linear time-invariant (LTI) systems with asymptotically stable and symmetric dynamics matrices.

In [10], the authors focus on *leader-follower linearly diffusively coupled multi-agent systems* with agents having single-integrator dynamics. In particular, these systems have Laplacian-based dynamics, which means they are not asymptotically stable. The authors demonstrate how using partitions for model order reduction can be transformed to using Petrov-Galerkin projections, while preserving network structure and consensus in the reduced order model (ROM). Further, they derive a simple expression for the relative $\mathcal{H}_2$-error when using an *almost equitable partition* (AEP) and establish a lower bound based on AEPs when using any partition.

The authors of [11] study *networks of identical passive systems over weighted and directed graphs*, but confine to interconnections with tree structures. Otherwise, these systems would be more general then those in [10]. They present a clustering method relying on the analysis of the corresponding *edge system* to find adjacent subsystems to cluster. Furthermore, they prove that this method preserves the consensus property. Nonetheless, clustering only adjacent subsystems might be restrictive.

The paper [12] presents an efficient clustering-based method for $\mathcal{H}_2$ model order reduction of *positive networks*, which include systems with Laplacian-based dynamics. Similarly to [9], the method is based on θ-reducible clusters and an $\mathcal{H}_2$-error bound. However, it is not clear how the ROMs resulting from this method compare to the $\mathcal{H}_2$-optimal ones. To the best of our knowledge, there is no efficient method for finding an $\mathcal{H}_2$-optimal ROM using graph partitions. Here, we propose an efficient $\mathcal{H}_2$-suboptimal method, originating from the $\mathcal{H}_2$-optimal model order reduction problem for LTI systems, and illustrate it on an example. Results indicate that the method finds a partition close to the optimal.

The outline of the paper is as follows. Section II explains notation and concepts used later. In Section III, we motivate and describe our model order reduction method, together with the issue of computing the $\mathcal{H}_2$-error for systems with Laplacian-based dynamics. We illustrate the method on an example in Section IV and conclude with Section V.

II. Preliminaries

Here we describe necessary notions for understanding the following sections. For details see [10].

A. Multi-Agent Systems

Let $G = (V, E, A)$ be a *weighted graph* with a *vertex set* $V = \{1, 2, \ldots, n\}$, $n \in \mathbb{N}$, an *edge set* E, and an *adjacency matrix* $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ with nonnegative elements. We say that

- G is *undirected* if $E \subseteq \{\{i, j\} : i, j \in V\}$,
- G is *directed* if $E \subseteq \{(i, j) : i, j \in V\}$,
- $a_{ij} > 0$ if and only if $\{i, j\} \in E$ or $(j, i) \in E$,
- G is *simple* if $a_{ii} = 0$ for all vertices $i \in V$,
- G is *connected* if for all vertices $i, j \in V$, $i \neq j$, there exist some vertices $i_1, i_2, \ldots, i_k \in V$, $k \geqslant 0$, such that $a_{i_1,i}, a_{i_2,i_1}, \ldots, a_{j,i_k}$ are positive,
- G is *symmetric* if $a_{ij} > 0 \Leftrightarrow a_{ji} > 0$ for all vertices $i, j \in V$.

[1]P. Mlinarić, S. Grundel, and P. Benner are with Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany. E-mails: mlinaric@mpi-magdeburg.mpg.de, grundel@mpi-magdeburg.mpg.de, benner@mpi-magdeburg.mpg.de.

[2]P. Mlinarić is also affiliated to the "International Max Planck Research School (IMPRS) for Advanced Methods in Process and System Engineering (Magdeburg)".

For undirected graphs we always have $A^T = A$, whereas for directed symmetric graphs this is not necessarily true.

We further define:

- *in-degree* $d_i = \sum_{j=1}^{n} a_{ij}$ of a vertex $i \in V$,
- *in-degree matrix* $D = \operatorname{diag}(d_1, d_2, \ldots, d_n)$,
- *Laplacian matrix* $L = D - A$,
- *incidence matrix* $R \in \mathbb{R}^{n\times p}$, where p is the number of edges ($p = |E|$) and every column r of R is associated to one edge $e \in E$ and satisfies

$$r_i = \begin{cases} 1, & \text{if } e = (i,j) \text{ for some } j \in V, \\ -1, & \text{if } e = (j,i) \text{ for some } j \in V, \\ 0, & \text{otherwise,} \end{cases}$$

  if G is directed, while e is given an arbitrary orientation if G is undirected,
- *edge-weights matrix* $W \in \mathbb{R}^{p\times p}$ with edge weights on its diagonal in the order corresponding to the incidence matrix R, and off-diagonal elements zero.

We now define a multi-agent system on a simple, undirected, weighted, and connected graph G. We declare some nodes to be *leaders* and others to be *followers*. Let $m \in \mathbb{N}$, $m \leqslant n$, be the number of leaders, $V_L = \{v_1, v_2, \ldots, v_m\} \subseteq V$ the set of leaders, and $V_F = V \setminus V_L$ the set of followers. Defining $M \in \mathbb{R}^{n\times m}$ with

$$M_{ij} = \begin{cases} 1, & \text{if } i = v_j, \\ 0, & \text{otherwise,} \end{cases}$$

the system

$$\dot{x}(t) = -Lx(t) + Mu(t), \tag{1a}$$
$$y(t) = W^{\frac{1}{2}} R^T x(t), \tag{1b}$$

is a *leader-follower linearly diffusively coupled multi-agent system*, where the agents have single-integrator dynamics, with input $u(t) \in \mathbb{R}^m$, state $x(t) \in \mathbb{R}^n$, and output $y(t) \in \mathbb{R}^p$.

The input $u(t)$ represents external input to leaders, the components of the state $x(t)$ are the values of the agents in the nodes, and the output $y(t)$ are weighted differences of the agents' values across edges. The property of interest for multi-agent systems is *reaching consensus*, which says that $x_i(t) - x_j(t) \to 0$ when $u \equiv 0$, for all $i, j \in V$ and all initial values. For system (1), this is equivalent to $y(t) \to 0$, because the multi-agent system is defined on a connected graph.

B. Model Order Reduction Preserving Structure and Consensus

Petrov-Galerkin projection is a general approach for reducing an arbitrary LTI system [13], [14]. On the example in (1), the approach consists of choosing two r-dimensional subspaces $\mathcal{V}_r, \mathcal{W}_r \subset \mathbb{C}^n$ and describing the dynamics of the ROM with:

Find $v(t) \in \mathcal{V}_r$ such that

$$\dot{v}(t) + Lv(t) - Mu(t) \perp \mathcal{W}_r.$$

Then the output is $\widehat{y}(t) = W^{\frac{1}{2}} R^T v(t)$.

By choosing some matrices $V_r, W_r \in \mathbb{C}^{n\times r}$ such that $\operatorname{Im} V_r = \mathcal{V}_r$ and $\operatorname{Im} \overline{W_r} = \mathcal{W}_r$ (where Im denotes the image of a matrix), we get the ROM

$$W_r^T V_r \dot{\widehat{x}}(t) = -W_r^T L V_r \widehat{x}(t) + W_r^T M u(t), \tag{2a}$$
$$\widehat{y}(t) = W^{\frac{1}{2}} R^T V_r \widehat{x}(t), \tag{2b}$$

which is an LTI system with input $u(t) \in \mathbb{R}^m$ (same as in (1)), state $\widehat{x}(t) \in \mathbb{R}^r$, and output $\widehat{y}(t) \in \mathbb{R}^p$.

In [10], the authors propose using

$$V_r = P(\pi), \tag{3a}$$
$$W_r = P(\pi)\left(P(\pi)^T P(\pi)\right)^{-1}, \tag{3b}$$

where $P(\pi)$ is a characteristic matrix of a partition π of the vertex set V. The characteristic matrix of a partition $\pi = \{C_1, C_2, \ldots, C_r\}$ is the matrix $P(\pi) \in \mathbb{R}^{n\times r}$ with

$$[P(\pi)]_{ij} = \begin{cases} 1, & \text{if } i \in C_j, \\ 0, & \text{otherwise.} \end{cases}$$

The ROM is then

$$\begin{aligned} \dot{\widehat{x}}(t) = &-\left(P(\pi)^T P(\pi)\right)^{-1} P(\pi)^T L P(\pi) \widehat{x}(t) \\ &+ \left(P(\pi)^T P(\pi)\right)^{-1} P(\pi)^T M u(t), \end{aligned} \tag{4a}$$
$$\widehat{y}(t) = W^{\frac{1}{2}} R^T P(\pi) \widehat{x}(t). \tag{4b}$$

Using (3) preserves structure, in the sense that system (4) represents a multi-agent system defined on a *directed, symmetric graph* [10]. Furthermore, using (3) preserves consensus, see [10, Theorem 4].

C. $\mathcal{H}_2$-Optimal Model Order Reduction Using Graph Partitions

The $\mathcal{H}_2$-norm is defined, for any $H \in \mathcal{H}_2$ (e.g. stable, strictly proper, rational matrix functions), by

$$\|H\|_{\mathcal{H}_2}^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} \|H(i\omega)\|_F^2 \, \mathrm{d}\omega,$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Let

$$H(s) = W^{\frac{1}{2}} R^T (sI + L)^{-1} M,$$
$$\widehat{H}(s) = W^{\frac{1}{2}} R^T V_r \left(sW_r^T V_r + W_r^T L V_r\right)^{-1} W_r^T M$$

be the transfer functions of systems (1) and (2), respectively. We consider the following $\mathcal{H}_2$-optimal model order reduction problem:

$$\min_{V_r, W_r \in \mathbb{R}^{n\times r}} \quad \|H - \widehat{H}\|_{\mathcal{H}_2}, \tag{5a}$$
$$\text{s.t.} \quad V_r = P(\pi), \tag{5b}$$
$$W_r = P(\pi)\left(P(\pi)^T P(\pi)\right)^{-1}, \tag{5c}$$
$$\pi \in \Pi, \ |\pi| = r, \tag{5d}$$

where Π is a set of all partitions of the vertex set V.

Problem (5) is actually a discrete optimization problem over a set of partitions of a set $\{1, 2, \ldots, n\}$ with r cells.

The number of such partitions is given by the Stirling number of the second kind $S(n, r)$ [15].

In the literature (to our best knowledge), there is no efficient method to exactly solve the optimization problem (5). Thus, our idea is to solve a relaxed problem, and use that solution to find a feasible solution for (5). We hope that this feasible solution is then close to the optimal solution as it is close to the optimal solution of the relaxed problem.

III. Method and $\mathcal{H}_2$-error computation

In this section, we first present our new method to find a partition in order to create a good ROM. Since numerically computing the $\mathcal{H}_2$-error is a crucial part in comparing our results, we dedicate the second half of this section to explaining how we did this.

A. The proposed method

1) Relaxation: We relax the problem (5) by dropping all constraints (5b), (5c), and (5d). Thus, we obtain the $\mathcal{H}_2$-optimal model order reduction problem for an LTI system. The *Iterative Rational Krylov Algorithm* (IRKA) finds a (locally) optimal solution efficiently for large-scale problems [14]. Note that the dynamics matrix in (1a) is not asymptotically stable, but IRKA can still be used because the transfer function of the system (1) is an element of $\mathcal{H}_2$. The fact that the transfer function associated to the LTI system (1) is indeed contained in $\mathcal{H}_2$ follows from the proof of Theorem 6 in [10]. See also Section III-B below for some elaboration on the technicalities associated to this problem.

IRKA will not in general solve the original problem (5). Therefore, IRKA will return as a result matrices V_r and W_r which will not (in general) satisfy (3).

2) Finding a Feasible Solution: In Petrov-Galerkin projection, the subspaces $\mathcal{V}_r$ and $\mathcal{W}_r$ are enough to determine the transfer function of the ROM. Therefore, the conditions (5b) and (5c) can be replaced by

$$\operatorname{Im} V_r = \operatorname{Im} P(\pi), \tag{6}$$

$$\operatorname{Im} W_r = \operatorname{Im}\Big(P(\pi)\left(P(\pi)^T P(\pi)\right)^{-1}\Big), \tag{7}$$

without changing $\widehat{H}$ and the cost (5a). The expression (6) motivates us to look for a partition π such that the image of $P(\pi)$ is "close" to the image of V_r obtained by IRKA.

We know that the condition (6) is equivalent to the existence of a nonsingular matrix Z such that $V_r = P(\pi)Z$. In general, condition (6) will not be satisfied, so there will be an error E such that $V_r = P(\pi)Z + E$. A very similar problem of finding a partition π was encountered in [16, §3], where a proposed solution is a clustering algorithm based on the *QR decomposition with column pivoting*. Algorithm 1 outlines the procedure. The idea behind it is the claim of the following Lemma, which says that if $V_r = P(\pi)Z$, the procedure will return the correct result.

Lemma 1: Algorithm 1 returns π with $P(\pi)Z$ as input, for an arbitrary partition π and a nonsingular matrix Z.

Proof: Let the number of cells in π be r, i.e. $|\pi| = r$. Let us denote the rows of Z with $z_1^T, z_2^T, \ldots, z_r^T$. Furthermore, without loss of generality we can assume that $P(\pi) = \operatorname{diag}\big(\mathbb{1}_{|C_1|}, \mathbb{1}_{|C_2|}, \ldots, \mathbb{1}_{|C_r|}\big)$, where $\mathbb{1}_k \in \mathbb{R}^k$ is a vector of all ones (this structure can be achieved by relabeling the vertices of the graph). Then we have

$$Z^T P(\pi)^T = \begin{bmatrix} z_1 \mathbb{1}_{|C_1|}^T & z_2 \mathbb{1}_{|C_2|}^T & \cdots & z_r \mathbb{1}_{|C_r|}^T \end{bmatrix}. \tag{8}$$

Therefore, $Z^T P(\pi)^T$ has repeating columns in blocks. If we perform the QR decomposition with column pivoting on (8) (ignoring the orthogonal matrix) and then undo the permutation of columns, the result is

$$\begin{bmatrix} \widetilde{z}_1 \mathbb{1}_{|C_1|}^T & \widetilde{z}_2 \mathbb{1}_{|C_2|}^T & \cdots & \widetilde{z}_r \mathbb{1}_{|C_r|}^T \end{bmatrix}, \tag{9}$$

where a permutation of the columns of $\begin{bmatrix} \widetilde{z}_1 & \widetilde{z}_2 & \cdots & \widetilde{z}_r \end{bmatrix}$ gives the upper-triangular matrix R_{11} from Algorithm 1. We see that multiplying (9) on the left by R_{11}^{-1} ($P(\pi)Z$ having full rank implies that R_{11} is nonsingular) and transposing produces $P(\pi)$, possibly with permuted columns. Thus we conclude that Algorithm 1 returns the partition π. ∎

Algorithm 1 Finding a partition using QR decomposition with column pivoting [16, §3]

Input: Matrix $V_r \in \mathbb{R}^{n\times r}$ of rank r
Output: Partition π such that $\operatorname{Im} P(\pi) \approx \operatorname{Im} V_r$

1: Perform QR decomposition with column pivoting on the matrix V_r^T, i.e. find orthogonal $Q \in \mathbb{R}^{r\times r}$, upper-triangular $R \in \mathbb{R}^{r\times n}$, and a permutation matrix $P \in \mathbb{R}^{n\times n}$ such that $V_r^T P = QR$
2: Partition R as $\begin{bmatrix} R_{11} & R_{12} \end{bmatrix}$, with $R_{11} \in \mathbb{R}^{r\times r}$ square upper-triangular and $R_{12} \in \mathbb{R}^{r\times(n-r)}$
3: Solve the triangular system $R_{11}X = R_{12}$
4: Compute $Y = P\begin{bmatrix} I_r & X \end{bmatrix}^T = [y_{ij}] \in \mathbb{R}^{n\times r}$
5: Find a partition $\pi = \{C_1, C_2, \ldots, C_r\}$ such that $i \in C_j$ if and only if $j = \arg\max_k |y_{ik}|$
6: Return π

The following Lemma gives the time and space complexity of Algorithm 1, proving that it is efficient even in the large-scale setting.

Lemma 2: In Algorithm 1, the number of floating point operations is $\mathcal{O}(nr^2)$ and the size of additional storage is $\mathcal{O}(nr)$.

Proof: The QR decomposition with column pivoting in line 1 performs r iterations. Each iteration consists of:

- looking for a (sub)column with maximum Euclidean norm (there are $\mathcal{O}(n)$ columns and computing the norm takes $\mathcal{O}(r)$ flops),
- replacing two columns if necessary ($\mathcal{O}(r)$ swaps), and
- applying a Householder reflection on a (sub)matrix of size $\mathcal{O}(r) \times \mathcal{O}(n)$ ($\mathcal{O}(nr)$ flops).

It follows that the number of operations in line 1 is $\mathcal{O}(nr^2)$. The amount of necessary storage is $nr + n$ for the matrix R and the permutation matrix P.

In line 3 it is necessary to solve a triangular system of size r with $n - r$ right-hand sides. Solving one triangular system takes $\mathcal{O}(r^2)$ flops. Therefore, it takes $\mathcal{O}(nr^2)$ flops to solve a system in line 3, without necessary additional storage—the solution can be stored in the place of R_{12}.

The calculation in line 4 does not need to be performed explicitly and can be included in the operation in the next line. In line 5 it is necessary to find maximum elements in n vectors of size r, which can be done in $\mathcal{O}(nr)$ comparisons.

Adding the contributions of all the lines in Algorithm 1 gives us the claim of the Lemma. ■

B. Computing the $\mathcal{H}_2$-error

When using an AEP, the relative $\mathcal{H}_2$-error can be computed directly [10, Theorem 6]. In other cases, we need to solve Lyapunov equations to compute the $\mathcal{H}_2$-norms.

To use a Lyapunov equation to compute the $\mathcal{H}_2$-norm, it is necessary that the state dynamics matrix is stable. The problem here is that neither the full system (1) nor the ROM (4) satisfy this, as the following Lemma explains.

Lemma 3: The state dynamics matrices of the systems (1) and (4) have nonpositive eigenvalues and their kernels are spanned by $\mathbb{1}$ (a vector of all ones of appropriate size).

Proof: The matrices in question are

$$-L \quad \text{and} \quad -\left(P(\pi)^T P(\pi)\right)^{-1} P(\pi)^T L P(\pi).$$

The claim for $-L$ follows from the properties of the Laplacian matrix of a connected graph. In particular, from positive semi-definiteness of L it follows that the eigenvalues of $-L$ are nonpositive. Further, from the definition of the Laplacian matrix, we see that $\mathbb{1}$ is in the kernel of $-L$. Finally, from the assumption of connectedness of the graph, we conclude that the kernel of $-L$ is a one-dimensional subspace.

For the second matrix, from [10, Lemma 3] we can conclude that its eigenvalues are nonpositive. Lastly, from [10, Theorem 4] it follows that $\mathbb{1}$ spans its kernel. ■

The following Lemma shows that both the full system (1) and the ROM (4) have transfer functions from $\mathcal{H}_2$, even though their state dynamics matrices have zero eigenvalues.

Lemma 4: The state $\mathbb{1}$ is unobservable for systems (1) and (4).

Proof: From $R^T\mathbb{1} = 0$ and $P(\pi)\mathbb{1} = \mathbb{1}$, it follows that $\mathbb{1}$ is in the kernel of both output matrices of systems (1) and (4), which implies the claim. ■

Applying Lemmas 3 and 4, we use the state transformation $\widetilde{x} = Tx$ with

$$T = \begin{bmatrix} 1 & & & -1 \\ & \ddots & & \vdots \\ & & 1 & -1 \\ & & & 1 \end{bmatrix}, \; T^{-1} = \begin{bmatrix} 1 & & & 1 \\ & \ddots & & \vdots \\ & & 1 & 1 \\ & & & 1 \end{bmatrix}$$

and eliminate the last (unobservable) component of $\widetilde{x}$ to obtain an equivalent system with a stable state dynamics matrix.

IV. Example

We used the example from [10] (see Figure 1). It is a leader-follower linearly diffusively coupled multi-agent system, defined on a simple, undirected, weighted, connected graph. The graph has $\{1, 2, \ldots, 10\}$ as the vertex set and 15 edges. Vertices 6 and 7 are the leaders of the multi-agent system. The Laplacian and input matrices are

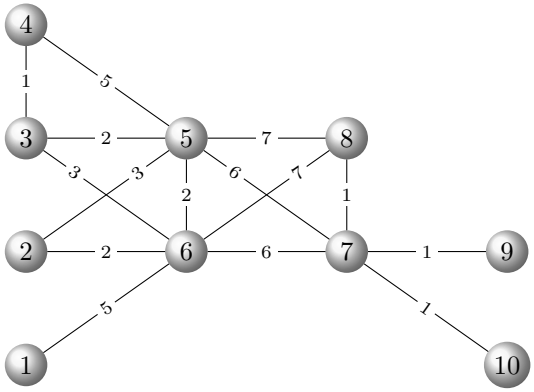

Fig. 1. Example, used in [10], of a leader-follower linearly diffusively coupled multi-agent system on a simple, undirected, weighted, connected graph. Vertices 6 and 7 are leaders.

$$L = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 & -5 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & -3 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6 & -1 & -2 & -3 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 6 & -5 & 0 & 0 & 0 & 0 & 0 \\ 0 & -3 & -2 & -5 & 25 & -2 & -6 & -7 & 0 & 0 \\ -5 & -2 & -3 & 0 & -2 & 25 & -6 & -7 & 0 & 0 \\ 0 & 0 & 0 & 0 & -6 & -6 & 15 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & -7 & -7 & -1 & 15 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix}, \quad M = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

We chose an edge ordering and orientation such that the incidence and edge-weights matrices are

$$R = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and $W = \operatorname{diag}(5, 3, 2, 1, 2, 3, 5, 2, 6, 7, 6, 7, 1, 1, 1)$.

The partition

$$\{\{1, 2, 3, 4\}, \{5, 6\}, \{7\}, \{8\}, \{9, 10\}\} \tag{10}$$

is an AEP of the graph in Figure 1 used in [10]. It has five cells, so we compare the relative $\mathcal{H}_2$-errors of different ROMs of order $r = 5$. The partition (10) is actually the only AEP with five cells, among the total of five AEPs of the graph in Figure 1.

From [10, Theorem 6], it follows that for the AEP in (10), the relative $\mathcal{H}_2$-error is

$$\frac{\|H - \widehat{H}_{\text{AEP}}\|_{\mathcal{H}_2}}{\|H\|_{\mathcal{H}_2}} = \sqrt{\frac{\left(1 - \frac{1}{2}\right) + \left(1 - \frac{1}{1}\right)}{2\left(1 - \frac{1}{10}\right)}} \approx 0.527046, \tag{11}$$

where $\widehat{H}_{\text{AEP}}$ is the transfer function of the ROM using the graph partition (10).

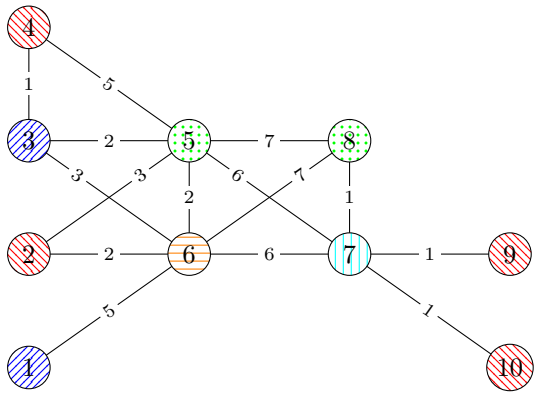

Fig. 2. Graph partition (13) with five cells. The assignment of vertices to cells is represented with different patterns and colors.

Next, we used IRKA to find a ROM of order $r = 5$. It found a ROM with relative $\mathcal{H}_2$-error of

$$\frac{\|H - \widehat{H}_{\text{IRKA}}\|_{\mathcal{H}_2}}{\|H\|_{\mathcal{H}_2}} \approx 0.0330412, \tag{12}$$

which is almost 16 times better than (11).

IRKA returned

$$V_r \approx \begin{bmatrix} -0.129 & -0.332 & -0.234 & 0.377 & 0.470 \\ -0.053 & -0.305 & -0.264 & -0.114 & 0.219 \\ -0.077 & -0.307 & -0.250 & 0.019 & 0.280 \\ -0.005 & -0.294 & -0.289 & -0.411 & 0.075 \\ -0.023 & -0.346 & -0.069 & 0.154 & -0.645 \\ -0.927 & -0.044 & 0.331 & -0.169 & 0.010 \\ 0.310 & -0.561 & 0.747 & -0.087 & 0.151 \\ -0.138 & -0.324 & -0.088 & 0.555 & -0.334 \\ 0.010 & -0.194 & -0.158 & -0.393 & -0.220 \\ 0.010 & -0.194 & -0.158 & -0.393 & -0.220 \end{bmatrix}.$$

The partition resulting from Algorithm 1 applied to the above matrix is

$$\{\{1,3\},\{2,4,9,10\},\{5,8\},\{6\},\{7\}\}, \tag{13}$$

and is shown in Figure 2. The relative $\mathcal{H}_2$-error of a ROM using partition (13) is

$$\frac{\|H - \widehat{H}_{V_r}\|_{\mathcal{H}_2}}{\|H\|_{\mathcal{H}_2}} \approx 0.150654, \tag{14}$$

which is more than 4 times worse than (12), but 3 times better than (11).

We notice by (7) that W_r can also be used to find a good partition. IRKA found

$$W_r \approx \begin{bmatrix} -0.211 & -0.372 & -0.034 & 0.295 & 0.107 \\ -0.068 & -0.160 & -0.343 & 0.064 & 0.075 \\ -0.121 & -0.172 & -0.198 & 0.178 & 0.112 \\ 0.032 & -0.077 & -0.592 & -0.128 & 0.027 \\ 0.070 & -0.117 & 0.062 & -0.344 & -0.857 \\ 0.868 & -0.175 & 0.247 & -0.011 & 0.235 \\ -0.336 & -0.040 & 0.335 & -0.714 & 0.403 \\ -0.252 & -0.108 & 0.56 & 0.461 & -0.128 \\ 0.009 & 0.610 & -0.019 & 0.099 & 0.013 \\ 0.009 & 0.610 & -0.019 & 0.099 & 0.013 \end{bmatrix},$$

TABLE I

TOP 20 PARTITIONS WITH 5 CELLS FOR REDUCING THE MULTI-AGENT SYSTEM IN FIGURE 1

Rank	Relative $\mathcal{H}_2$-error	Partition
1	0.128053	$\{\{1,8\},\{2,3,4,9,10\},\{5\},\{6\},\{7\}\}$
2	0.131311	$\{\{1,2,3,4\},\{5,8\},\{6\},\{7\},\{9,10\}\}$
3	0.137466	$\{\{1,2,3,4,9,10\},\{5\},\{6\},\{7\},\{8\}\}$
4	0.137473	$\{\{1,3,8\},\{2,4,9,10\},\{5\},\{6\},\{7\}\}$
5	0.143700	$\{\{1,5,8\},\{2,3,4\},\{6\},\{7\},\{9,10\}\}$
6	0.145900	$\{\{1,2,3\},\{4,9,10\},\{5,8\},\{6\},\{7\}\}$
7	0.146196	$\{\{1,8\},\{2,3,4,9\},\{5,10\},\{6\},\{7\}\}$
8	0.146196	$\{\{1,8\},\{2,3,4,10\},\{5,9\},\{6\},\{7\}\}$
9	0.147022	$\{\{1,2,3,8\},\{4,9,10\},\{5\},\{6\},\{7\}\}$
10	0.149240	$\{\{1,8,9\},\{2,3,4,10\},\{5\},\{6\},\{7\}\}$
11	0.149240	$\{\{1,8,10\},\{2,3,4,9\},\{5\},\{6\},\{7\}\}$
12	0.149654	$\{\{1,8\},\{2,4,9,10\},\{3,5\},\{6\},\{7\}\}$
13	0.150440	$\{\{1,5\},\{2,3,4,9,10\},\{6\},\{7\},\{8\}\}$
14	0.150654	$\{\{1,3\},\{2,4,9,10\},\{5,8\},\{6\},\{7\}\}$
15	0.151684	$\{\{1,2,8\},\{3,4,9,10\},\{5\},\{6\},\{7\}\}$
16	0.153100	$\{\{1,2,3,4,9\},\{5,8\},\{6\},\{7\},\{10\}\}$
17	0.153100	$\{\{1,2,3,4,10\},\{5,8\},\{6\},\{7\},\{9\}\}$
18	0.153819	$\{\{1\},\{2,3,4,9,10\},\{5,8\},\{6\},\{7\}\}$
19	0.154374	$\{\{1,3,8,9\},\{2,4,10\},\{5\},\{6\},\{7\}\}$
20	0.154374	$\{\{1,3,8,10\},\{2,4,9\},\{5\},\{6\},\{7\}\}$

and in this example, Algorithm 1 returns the partition

$$\{\{1,2,3,9,10\},\{4,8\},\{5\},\{6\},\{7\}\}. \tag{15}$$

The relative $\mathcal{H}_2$-error when using the partition (15) is

$$\frac{\|H - \widehat{H}_{W_r}\|_{\mathcal{H}_2}}{\|H\|_{\mathcal{H}_2}} \approx 0.179746,$$

which is worse than (14).

It is also possible to use IRKA's V_r and W_r together. By trial and error, we found that using $V_r - \frac{1}{2}W_r$ produces the partition

$$\{\{1,2,3\},\{4,9,10\},\{5,8\},\{6\},\{7\}\}, \tag{16}$$

which results in the relative $\mathcal{H}_2$-error

$$\frac{\|H - \widehat{H}_{V_r - \frac{1}{2}W_r}\|_{\mathcal{H}_2}}{\|H\|_{\mathcal{H}_2}} \approx 0.1459.$$

It is however not clear how to pick such a linear combination and further research is needed to optimize this method.

Table I shows the best 20 partitions of size 5 and their relative $\mathcal{H}_2$-errors. It should be noted that the total number of partitions of a set with 10 elements into 5 cells is $S(10,5) = 42\,525$. We notice that partition (13) is the 14th partition and that the best partition produces about 1.18 times better error. Additionally, partition (16) is the 6th in Table I.

Table I does not show that partition (15) is 192nd and partition (10) is 5996th. Thus, the new method gets a lot closer to the optimal partition than the AEP in this example.

Furthermore, from Table I we see that, in all of the top 20 partitions, leaders 6 and 7 appear in singletons. This makes sense, because this way no input is diffused over more agents. However, further research is needed to see if, in general, the best partition has leaders appearing in singletons.

As a final remark, the method we presented here can be straightforwardly generalized. Just from the main ingredients, IRKA and QR decomposition, it is clear that the method

can be applied to any LTI system with a transfer function in $\mathcal{H}_2$. This includes, e.g., systems defined over directed graphs. Next, if another cost functional is preferred over the $\mathcal{H}_2$-error, a different model order reduction method (within the Petrov-Galerkin projection framework) can be used. Finally, we do not claim that clustering via QR decomposition is in any way optimal. It is possible to use any of the other clustering methods, but further research is needed to see which one would be, e.g., $\mathcal{H}_2$-optimal.

V. Conclusions

We presented our method, combining IRKA and a clustering algorithm, for model order reduction of multi-agent systems using graph partitions. It seems heuristically that this method is able to create a partition whose $\mathcal{H}_2$-error is small. We furthermore elaborated that this method is scalable to large-scale systems. Our numerical test for a small network shows that, among $42\,525$ partitions, our algorithm found the 14th best approximation whose error is of the same order of magnitude as the error of the best partition. A theoretical foundation that the algorithm always finds a partition in some sense close to optimal remains an open problem for future work. Ultimately, one would of course like to solve the constrained optimization problem (5) with low complexity, allowing to apply such an algorithm to large-scale problems.

References

[1] M. E. J. Newman, *Networks: An Introduction*. New York, NY, USA: Oxford University Press, Inc., Mar. 2010.

[2] E. Estrada, M. Fox, D. J. Higham, and G.-L. Oppo, Eds., *Network Science: Complexity in Nature and Technology*. London, UK: Springer-Verlag, 2010.

[3] P. Benner, R. Findeisen, D. Flockerzi, U. Reichl, and K. Sundmacher, Eds., *Large-Scale Networks in Engineering and Life Sciences*, ser. Modeling and Simulation in Science, Engineering and Technology. Basel, CH: Birkhäuser, 2014.

[4] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[5] S. Martini, M. Egerstedt, and A. Bicchi, "Controllability decompositions of networked systems through quotient graphs," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, Dec. 2008, pp. 5244–5249.

[6] A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt, "Controllability of multi-agent systems from a graph-theoretic perspective," *SIAM J. Cont. Optim.*, vol. 48, no. 1, pp. 162–186, 2009.

[7] S. Martini, M. Egerstedt, and A. Bicchi, "Controllability analysis of multi-agent systems using relaxed equitable partitions," *Int. J. Syst., Control Commun.*, vol. 2, no. 1/2/3, pp. 100–121, Jan. 2010.

[8] A. Chapman and M. Mesbahi, "UAV flocking with wind gusts: Adaptive topology and model reduction," in *Proceedings of the American Control Conference*, June 2011, pp. 1045–1050.

[9] T. Ishizaki, K. Kashima, J.-I. Imura, and K. Aihara, "Model reduction and clusterization of large-scale bidirectional networks," *IEEE Trans. Autom. Control*, vol. 59, no. 1, pp. 48–63, Jan. 2014.

[10] N. Monshizadeh, H. L. Trentelman, and M. K. Çamlibel, "Projection-based model reduction of multi-agent systems using graph partitions," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 2, pp. 145–154, June 2014.

[11] B. Besselink, H. Sandberg, and K. H. Johansson, "Model reduction of networked passive systems through clustering," in *Proc. European Control Conf. ECC 2014, Strasbourg*, June 2014, pp. 1069–1074.

[12] T. Ishizaki, K. Kashima, A. Girard, J.-I. Imura, L. Chen, and K. Aihara, "Clustering-based $\mathcal{H}_2$-state aggregation of positive networks and its application to reduction of chemical master equations," in *51st IEEE Conference on Decision and Control (CDC), Maui, Hawaii*, Dec. 2012, pp. 4175–4180.

[13] P. Benner, "Numerical linear algebra for model reduction in control and simulation," *GAMM-Mitt.*, vol. 29, no. 2, pp. 275–296, 2006.

[14] A. C. Antoulas, C. A. Beattie, and S. Gugercin, "Interpolatory model reduction of large-scale dynamical systems," in *Efficient Modeling and Control of Large-Scale Systems*, J. Mohammadpour and K. M. Grigoriadis, Eds. Springer US, 2010, pp. 3–58.

[15] B. C. Rennie and A. J. Dobson, "On Stirling numbers of the second kind," *Journal of Combinatorial Theory*, vol. 7, no. 2, pp. 116–121, 1969. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021980069800451

[16] H. Zha, X. He, C. Ding, H. Simon, and M. Gu, "Spectral relaxation for k-means clustering," in *NIPS*, 2001, pp. 1057–1064.