Model Reduction of Power Grids via Clustering

Author: Joris J. Berendschot

First supervisor: Prof. dr. C. de Persis

Second supervisor: Dr. A. J. Bosch

Daily supervisors: H.J. van Waarde M. Jeeninga

Industrial Engineering and Management
Production Technology and Logistics

January 23, 2019
Abstract

In this project, we are concerned with strategies for the clustering of nodes based on the cluster-based model reduction for power grid networks. To minimize the number of failures in these networks, it is important to know the behaviour. To predict the behaviour of these networks, the size and complexity of the network can be reduced with a model reduction technique. Based on the cluster-based model reduction technique we attempt to find a strategy to cluster nodes. At this stage in the research, analysing a power grid is too big and complex and therefore a toy model of ten nodes is analysed. In the attempt to find a strategy, we want to find patterns in the optimal partitions of structured graphs, random graphs and weighted graphs. We considered two different cases: in case 1 the error is measured considering the weighted differences of the states of neighbouring nodes, and for case 2 the error is measured considering the weighted differences of the states of all other nodes. The error calculations are based on the $H_2$-norm.
# Contents

Abstract 3

Introduction 7

1 Problem analysis 9
  1.1 Problem description .................................................. 9
    1.1.1 System Description ............................................. 9
    1.1.2 Size reduction of systems .................................... 10
    1.1.3 Model reduction techniques ..................................... 10
    1.1.4 System specification ........................................... 10
    1.1.5 Cluster-based model reduction technique ...................... 11
    1.1.6 Error calculation ............................................... 11
  1.2 Problem statement .................................................. 11
  1.3 Research method .................................................... 12

2 Cluster-based model reduction 13
  2.1 Preliminaries .......................................................... 13
    2.1.1 Systems .......................................................... 13
    2.1.2 Graph theory .................................................... 13
    2.1.3 State of the nodes ............................................. 14
    2.1.4 Full matrix $\bar{R}$ ........................................... 15
    2.1.5 $H_2$-norm ....................................................... 15
  2.2 Reduced order system .............................................. 15
    2.2.1 Partition matrix $P$ .......................................... 16
    2.2.2 Reduced order matrices ...................................... 16
    2.2.3 Extended matrices ............................................ 16
  2.3 Error calculation .................................................... 17
    2.3.1 Lyapunov method ............................................... 17
    2.3.2 Lyapunov command in MATLAB .................................. 17
  2.4 MATLAB calculations .............................................. 18
    2.4.1 Input matrices ................................................ 18
    2.4.2 Clustering program ............................................ 18
    2.4.3 Analysed graphs ............................................... 18

3 Results 19
  3.1 Case 1: incidence matrix $R$ ...................................... 19
    3.1.1 Structured graphs ............................................. 19
    3.1.2 Random graphs ................................................ 20
    3.1.3 Weighted graphs ............................................... 20
    3.1.4 Conclusions for case 1 ........................................ 20
    3.1.5 Strategy for case 1 ............................................ 21
  3.2 Case 2: full matrix $\bar{R}$ ...................................... 21
    3.2.1 Structured graphs ............................................. 21
    3.2.2 Random graphs ................................................ 21
    3.2.3 Weighted graphs ............................................... 22
    3.2.4 Conclusions for case 2 ........................................ 22

Conclusion 23

Appendices 25
A  Manually created $R$, $M$ and $W$ matrix  
B  Random $R$ matrix generator  
C  $P$ matrix generator  
D  Clustering program  
E  Results case 1: incidence matrix $R$ 
   E.1 Structured graphs of case 1  
   E.2 Random graphs of case 1  
   E.3 Weighted graphs of case 1  
F  Results case 2: full matrix $\bar{R}$ 
   F.1 Structured graphs of case 2  
   F.2 Random graphs of case 2  
   F.3 Weighted graphs of case 2
Introduction

The modern day society depends greatly on a steady supply of energy. A stable power grid network is therefore very important. Small breakdowns in the supply of electricity can have big influences on many different sectors, such as hospitals, airports, harbours, among many others. The power grid includes many actors, such as different electricity producers, electricity distributors, consumers and electricity distribution lines. If the behaviour of the system is known, the number of failures in the power grid network can be reduced. Mathematical models can be used to analyse and predict the behaviour of the power grids. Nevertheless, due to computational limitations it is hard to execute an analysis on big and complex networks. By reducing the size and complexity of the network, prediction of the network becomes easier and faster.

There are several methods to reduce the size and complexity of a network, and one of these methods is the cluster-based model reduction. To use this method, a network is represented by a model with nodes and edges. Nodes represent actors such as the distributors and consumers, and edges are distribution lines. The cluster-based model reduction technique has the ability to find a reduced order model given a partitions of nodes. Each division of nodes over a number of clusters is called a partition. When a system is reduced in size via the cluster-based model reduction technique, the network topology of the original system is similar to the reduced order model’s network topology [Monshizadeh, Trentelman, and Camlibel, 2014]. This implies that the structure of the reduced order system is similar to the original system. For power grid networks, the topology is important and therefore we use the cluster-based model reduction technique.

The reduced order model is only valuable for analysis if it behaves in a similar way as the original network. To determine how good the reduced order model approximates the behaviour of the original network, we compute an error calculation based on the $H_2$-norm [Cheng and Scherpen, 2016]. The smaller the error, the better the reduced order model approximates the behaviour of the original network.

There are many different possibilities to divide a given number of nodes into a number of clusters. Based on the Stirling numbers of the second kind, a network with ten nodes that is divided into five clusters, has a total of 42525 partitions [Joarder and Mahmood, 1997]. For each partition the error can be calculated, and the partition with the smallest error is the optimal partition. Calculating the error for every possible partitions is a time-consuming task, especially for big networks such as power grids. If the practical interpretation of the influencing factors that lead to the partition with the smallest error is known, a clustering strategy can be created. With the strategy, a network can be reduced in size without the help of calculating the errors of all different possible partitions.

To find a strategy for the clustering of nodes, we compute and analyse the optimal partitions of many different graphs. Since analysing networks with the size of a power grid network is too difficult and complex for this moment, we will analyse networks of ten nodes. Different graphs with different structures and weights are analysed. The analysis is executed for two different cases. In the first case, the error is measured considering the weighted differences of the states of neighbouring nodes. In the second case, the weighted differences of the states of all other nodes in the system are considered. Eventually, we attempt to find strategies for the clustering of nodes for both cases.

The outline of this project is as follows. In chapter 1, we provide a more in depth analysis of the problem. In chapter 2, we explain the cluster-based model reduction technique. In chapter 3, the results are discussed and a strategy for the clustering of both cases is determined.
Chapter 1

Problem analysis

1.1 Problem description

The power grid is a system which transports electricity. It connects many different actors in the system, from the electricity producers to the end consumers including all the distributing stations in between. Failures in the supply of electricity have a big impact on the modern day society. Small breakdowns in the supply of electricity to for example manufacturing plants can lead to big costs, while in hospitals the consequences can be even worse. With mathematical models the behaviour of the networks can be analysed and predicted. With this, breakdowns and failures of the power grid system can be prevented. However, due to computational limitations it is difficult to execute the analysis on these big and complex networks.

1.1.1 System Description

Since there are many different ways to simplify a system, it is important to know the important aspects of the system. First of all, the actors inside the system are important. The main actors inside a power grid network can be defined by [Fan et al., 2014]:

Main electricity producers

The main producers generate the electricity that is distributed throughout the power grid. Some examples of main producers are coal plants, nuclear plants and hydro-electric plants.

Small producers

The small producers generate small amounts of electricity. Households that have solar panels are an example of small producers. This group generates relatively small amounts of electricity.

Electricity distributors

This group distributes the electricity that is generated by the main and small producers over all the consumers.

Consumers

All households, factories and everything else that consumes electricity are part of this group.

High voltage distribution lines

The transportation of the electricity from the main producers to the energy distributors happens through these lines.

Distribution lines

These lines are used by the electricity distributors to transport the electricity from the distribution centres to the consumers.
1.1.2 Size reduction of systems

A possible method to predict the behaviour of the original network is to conduct an analysis on a smaller and less complex network, which approximates the behaviour of the original network. In order to reduce the complexity and size of a system, the system can be transformed into a model containing nodes and edges. In the context of power grids, a node can be viewed upon as for example an electricity producer, an electricity distributor or a consumer. The edges can be viewed upon as distribution lines. On the model with nodes and edges we can apply model reduction techniques. There are many different model reduction techniques, and each method has its own advantages and disadvantages. To reduce the size and complexity of a model, either the mutual relationship between nodes and the structure of the graph can be simplified, or the complexity of the nodes itself can be reduced. The latter, the so-called ‘dynamic order’ of the nodes, is already researched by Monshizadeh [Monshizadeh, Trentelman, and Camlibel, 2013]. Therefore, we focus on reducing the complexity and size of the graphs structure, also called the network topology. The network topology is the combination of the arrangement of the nodes and the conductance of the edges. Consequently, it is important to preserve the network topology of the original model in the simplified model, which is also called a reduced order model.

1.1.3 Model reduction techniques

A common way to reduce the size and complexity of a model is by placing nodes into different clusters. With the division of nodes into clusters a network is created which contains less components. A reduced order model is valuable for analysis if it is a good approximation of the original model. In order to determine this we can calculate the error between the two systems. The smaller the error, the better the reduced order model approximates the behaviour of the original system.

There are multiple different techniques to reduce a model with the use of clusters, and each method has its own advantages and disadvantages. One of the most popular model reduction techniques is the balanced truncation method. With this method, it is possible to determine the optimal number of clusters corresponding to a demanded error in a fast and easy way. Nevertheless, this method has one big drawback: it destroys the network topology of the original model. Two other examples of commonly used model reduction techniques are the ‘Optimal Hankel norm approximation’ and the ‘moment matching technique’. Although these methods have the ability to generate a reduced-order model which approximates the behaviour of the original system, they are likely to destroy the network topology of the original network too [Besselink, Sandberg, and Johansson, 2016]. Therefore, the analysis of the reduced order system will not give relevant results for power grids.

Another method is the cluster-based model reduction technique. With this method, the network topology of the original systems is similar to the reduced order model’s network topology. This implies that the structure of the reduced order system is similar to the original system [Monshizadeh, Trentelman, and Camlibel, 2014]. It can be expected that the behaviour of the reduced order model is similar to the behaviour of the original model when analysing power grids. Therefore, we will use the cluster-based model reduction technique in this project.

1.1.4 System specification

In the sense of power grid networks, we define our system as an RC circuit. The RC circuit is a simplified ‘toy model’ of a power grid network. In the toy model the nodes represent conductors and the edges correspond to resistors. An RC circuit can be regarded as a multi-agent system. [Monshizadeh, Trentelman, and Camlibel, 2014]. The nodes can be divided into two different groups: followers and leaders. The leaders have the ability to receive input from outside the system. In this way, the state of the system can be influenced. The followers do not have these characteristics [Weiss, 2013].

In general graphs can have directed and undirected edges. A directed edge is an edge that has a one-way communication, whereas in undirected edges the communication occurs in both directions. For the case of power grids, communication can be interpreted as the distribution of power. For the simplification of the project we only focus on graphs which solely contain undirected edges, called undirected graphs. However, the reduced order network can be a directed graph. Moreover, we only focus on controllable graphs, not on uncontrollable graphs.
We will analyse simple graphs, which means that the graph does not contain self-loops and multiple edges between two nodes are not present. Moreover, the analysed graphs are static. This means that the edges that connect the nodes will not change position over time [Monshizadeh, Trentelman, and Camlibel, 2014].

The focus is on weighted graphs. Therefore, all edges have a certain weight attached to them. This weight represents conductance of an edge. In some cases, a graph will be called unweighted. In essence this is the same a weighted graph in which each edge has a weight of one. As described in the problem description, power grids are big and complex. Intuitively, big is over a thousand nodes. In order to obtain a first intuition on which strategy is followed in the clustering of nodes, a thousand nodes is too many. Therefore, we analyse smaller graphs containing ten nodes. The strategy that occurs in the clustering of graphs with ten nodes will most probably also be present in bigger graphs. Finding a strategy for bigger graphs is more complicated and, in this stage of the research, too difficult.

1.1.5 Cluster-based model reduction technique

In order to use the cluster-based model reduction technique, the weighted graph that is analysed is transformed into matrices. A weighted graph can be described by the combination of two matrices: the incidence matrix and the weight matrix. The incidence matrix contains information about which nodes are connected to each other. The weight matrix contains the conductance of each edge. With the incidence matrix and the weight matrix, the Laplacian matrix can be computed. The Laplacian matrix contains all the information about the graph. The cluster-based model reduction technique transforms the Laplacian matrix into a reduced order Laplacian matrix with the help of a partition matrix. The partition matrix describes which node is assigned to which cluster. Note that the reduced order Laplacian does not represent the original system which is based on nodes, but the reduced order model which is based on the clusters.

1.1.6 Error calculation

An error calculation of the reduced network, compared to the original network, can be computed. With the error, we can determine how good the reduced order model approximates the behaviour of the original system. The error can be computed in many different ways, and we choose to calculate it based on the $H_2$-norm. The $H_2$-norm has several advantages over other error calculation methods: the expression for the computation of the error is simple, easy to compute, and can be derived directly from the matrices that are used in the cluster-based model reduction technique [Monshizadeh, Trentelman, and Camlibel, 2014].

1.2 Problem statement

As stated before, as a tool to analyse large complex power grid networks, the cluster-based model reduction technique is the best way to reduce the size and complexity of a power grid network. The method has the ability to compute the error, given a certain number of clusters. With this we can determine the optimal partition. However, there is no practical interpretation of which factors influence a partition to have the lowest error. If there is a certain strategy in the clustering of nodes that works for every graph, the graphs can already be reduced in size before further analysis on the behaviour of the system is conducted. Hence, we focus on finding a strategy for the clustering of the nodes. This results in the following problem statement.

The cluster-based model reduction technique has the ability to obtain a reduced order model given a partition of nodes. Currently, there is no strategy to obtain the optimal partition for a given number of clusters.
1.3 Research method

To find a strategy to obtain the optimal partition for a given number of clusters, we must find out the optimal partitions of graphs. Given a number of clusters, there are multiple different partitions. We calculate the error for each partition, and we define the partitions with the lowest error as the optimal partition for that given number of clusters. The optimal partition is found for every possible number of clusters, and based on these optimal partitions we try to find patterns. After the analysis of multiple graphs, the patterns can be combined and a strategy can be constructed.

We want to find a strategy for the clustering of nodes for two different cases. In the first case, the error is measured considering the weighted differences of the states of neighbouring nodes. In the second case, the error is measured considering the weighted differences of the states of all other nodes. In the next chapter we will explain this in a more detailed way.
Chapter 2

Cluster-based model reduction

Cluster-based model reduction is not a new technique. The mathematical computation of this method is already known and worked out in several papers. This chapter will explain all the steps of the cluster-based model reduction in detail. Section 2.1 and 2.2 are based on a paper of Monshizadeh [Monshizadeh, Trentelman, and Camlibel, 2014].

2.1 Preliminaries

In this section we provide the preliminaries that are necessary to execute the research. This section describes a more specific definition of the analysed system, the basics of graph theory, the leader-follower principle is introduced, the full matrix is introduced and the equation of the $H_2$-norm is defined.

2.1.1 Systems

A networked system can be defined as the combination of two things: the graph which represents the topology of a system and the node dynamics. The topology of a networked system describes the relationship between the nodes of the system, whereas the node dynamics describes how each node behaves. Therefore, in order to describe the system completely, both the graph and node dynamics must be described.

2.1.2 Graph theory

In this project we only consider weighted undirected graphs. An example of such a weighted graph is given in Figure 2.1. A weighted undirected graph can be described by $G := (V, E, A)$. $V$ is the vertex set, or equivalently, the set of nodes. The vertex set is defined by $V := \{1, 2, ..., n\}$. The edge set $E$ contains all the edges that connect nodes. Each edge is connected to two nodes, on one end of the edge to node $i$ and on the other end to node $j$. Since self-loops are not allowed, $i \neq j$. The adjacency matrix $A$ encodes the structure of the graph. Each entry denotes the weight of the edge which connects node $i$ to node $j$. Therefore, the adjacency matrix $A$ is given by $A := [a_{ij}]$, with $a_{ij}$ being the weight.

In order to find the incidence matrix $R$ of graph $G$, an arbitrary direction is given to each edge. This step is executed on the graph of Figure 2.1. Furthermore, each edge is given an edge-index. This results in the new graph given by Figure 2.2. In this figure, the weights are left out, but they remain the same as in Figure 2.1.

In the incidence matrix $R$, $i$ is the $i^{th}$ node and $j$ is the $j^{th}$ edge. In order to create the incidence matrix $R$, $R$ is presented by $R := [r_{ij}]$ with

$$r_{ij} := \begin{cases} 
1 & \text{if edge } j \text{ is outgoing from node } i \\
-1 & \text{if edge } j \text{ is going into node } i \\
0 & \text{otherwise.}
\end{cases}$$
We define the weight matrix $W$ by $W := \text{diag}(w_1, w_2, \ldots, w_k)$, in which $w_1$ is the weight of the first edge, $w_2$ is the weight of the second edge, and so on. With the incidence matrix $R$ and the weight matrix $W$ the Laplacian matrix $L$ of the original graph can be computed by $L := RWR^T$.

### 2.1.3 State of the nodes

In this project, graphs with leaders and followers are considered. Recall that a graph is given by $G := (V, E, A)$, in which $V := \{1, 2, \ldots, n\}$. We introduce the set with all leaders $V_L$. This leaders set is defined by $V_L := \{v_1, v_2, \ldots, v_m\} \subseteq V$, in which $v_1$ is the index of the node that is the first leader, $v_2$ is the index of the node that is the second leader, and so forth. Therefore, the followers set $V_F$ is now given by $V_F := V \setminus V_L$. With the use of an example, if a system consists of ten nodes, of which the second, third and sixth node are leaders, the sets are as follows: $V = \{1, 2, \ldots, 10\}$, $V_L = \{2, 3, 6\}$, $V_F = \{1, 4, 5, 7, 8, 9, 10\}$. The leader-follower multi-agent system can be transformed into a dynamical system:

$$
\dot{x}_i := \begin{cases} 
    z_i, & \text{if } i \in V_F \\
    z_i + u_l, & \text{if } i \in V_L.
\end{cases}
$$

In this dynamical system $\dot{x}_i \in \mathbb{R}$ is the state of node $i$, $u_l \in \mathbb{R}$ is the externally applied input to node $v_l$ with $l$ being the $l^{th}$ leader, and $z_i \in \mathbb{R}$ states the relationship between the nodes and is called a coupling variable. The variable is a diffusive coupling between the states of node $i$ and its neighbours, and is given by

$$
z_i := \sum_{j=1}^{n} a_{ij}(x_j - x_i).$$

The matrix that describes whether a node is a leader or a follower is $M$. Matrix $M \in \mathbb{R}^{n \times m}$ is defined by

$$M_{il} := \begin{cases} 
    1, & \text{if } i = v_l \\
    0, & \text{otherwise}.
\end{cases}$$

An example of a network with weights and leaders is given in Figure 2.3, in which the red nodes are leaders and the blue nodes are followers.

Lastly, let all the states $x_i$ of each node $i$ be collected in the vector $x = [x_1, x_2, \ldots, x_n]^T$. Moreover, let all the external inputs $u_l$, applied to node $v_l$, be collected in the vector $u = [u_1, u_2, \ldots, u_m]^T$. To create the state space representation, the output of the system must be defined. The output matrix $C$, in which the error is measured considering the weighted differences of the states of
neighbouring nodes, is defined by $C := W^{1/2}R^T$. The whole system based on graph $G$ can be described by a state space representation:

\[
\dot{x} = -Lx + Mu \\
y = Cx
\]

where $L$ is the Laplacian matrix, $M$ is the leader-follower matrix, $W$ is the weight matrix and $R$ is the incidence matrix of $G$.

2.1.4 Full matrix $\bar{R}$

As explained in section 1.2, we discuss two different cases. In the first case, the error is measured considering the weighted differences of the states of neighbouring nodes. This is done by using the incidence matrix $R$ as a parameter to calculate the output matrix $C$. In the second case, the error is measured considering the weighted differences of the states of all other nodes. In order to investigate this, the output matrix $C$ does not depend on the incidence matrix $R$, but it depends on the full matrix $\bar{R}$. The matrix $\bar{R}$ is the incidence matrix of a complete graph. In a complete graph all the nodes are connected to each other. Therefore, the output considers the weighted differences of the states of all other nodes in the model. Hence, $\bar{C} = W^{1/2}\bar{R}^T$. In the remainder of this chapter, the output matrix will be denoted by $\bar{C}$.

2.1.5 $H_2$-norm

The $H_2$-norm of a single system represents the average effect that the input has on the output. Therefore, it can be used to normalize the error of the comparison between a system and its reduced system. The normalization of the error is necessary to compare the errors of different graphs with the same size. The squared $H_2$-norm, denoted by $\gamma$, is computed by the following formula:

\[
\gamma^2 := \text{trace}(M^T \int_0^\infty e^{-L^T t}C^T Ce^{-Lt} dt M).
\]

2.2 Reduced order system

The original system is now described entirely. The next step in the cluster-based model reduction technique is to divide the nodes in clusters, called partitioning. The partitioning is solely on the basis of the node index numbers. Therefore, the leader-follower principle is not taken into account in the partitioning of the nodes. Recall that $n$ is the number of nodes. Moreover, recall that the vertex set of graph $G$ is $V = \{1, 2, ..., n\}$. The vertex set $V$ is split up in different clusters. The number of clusters in which the set is split up is $r$. Similar to the previous section, this section is based on the article of Monshizadeh [Monshizadeh, Trentelman, and Camlibel, 2014].
2.2.1 Partition matrix $P$

For the division of the nodes in different clusters, it is important that each node is placed in exactly one cluster. Any nonempty subset of $V$ is called a cell of $V$. Furthermore, a collection of cells, given by $\pi = \{K_1, K_2, ..., K_r\}$ is a partition of $V$ if and only if $\bigcup K_i = V$ and $K_i \cap K_j = \emptyset$ whenever $i \neq j$. The first part of the condition ensures that the combination of all the nodes in the cluster $K$ is equal to the set of all the nodes $V$, and therefore every node is placed into a cluster. The second part of the condition ensures that a node is not in two different clusters. Currently, the vertex set $V$ is split up in $r$ cluster cells, namely: $K_1, K_2, ..., K_r$. Every cell $K \subseteq V$ is defined as an $n$-dimensional characteristic column vector $p(K)$ with

$$p_i(K) := \begin{cases} 1, & \text{if } i \in K \\ 0, & \text{otherwise.} \end{cases}$$

Hence, for the partition $\pi := \{K_1, K_2, ..., K_r\}$, the partition matrix $P$ is defined by

$$P(\pi) = [p(K_1) \ p(K_2) \ ... \ p(K_r)].$$

For example, the partition matrix $P$, with $V = \{1, 2, ..., 10\}$, $r=3$ and $\pi = \{\{1, 3, 8, 9, 10\}, \{2, 7\}, \{4, 5, 6\}\}$ is given by

$$P(\pi) = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}^T.$$

2.2.2 Reduced order matrices

In order to determine the error between the reduced order model and the original model, the reduced order matrices must be calculated. The reduced order incidence matrix, which is denoted by $\hat{R}$, is calculated by $\hat{R} = P^T R$. Furthermore, the reduced order Laplacian matrix, the reduced order leader-follower matrix and the reduced order output matrix should be calculated. These matrices are denoted by $\hat{L}$, $\hat{M}$ and $\hat{C}$ and are defined by

$$\hat{L} := (P^T P)^{-1} P^T L P \quad \quad \hat{M} := (P^T P)^{-1} P^T M \quad \quad \hat{C} := W^{1/2} (P^T R)^T.$$

2.2.3 Extended matrices

To compute the error of the reduced order model compared to the original model, we combine the Laplacian matrix of the original system and the reduced order system into one matrix. This is also executed for the leader-follower matrices and the output matrices. These matrices are called the extended matrices. The extended matrices are defined by

$$L_e := \begin{bmatrix} L & 0 \\ 0 & \hat{L} \end{bmatrix} \quad \quad M_e := \begin{bmatrix} M \\ \hat{M} \end{bmatrix} \quad \quad C_e := \begin{bmatrix} C & -\hat{C} \end{bmatrix}.$$

In these matrices, the subscript $e$ stands for extended.
2.3 Error calculation

With the extended matrices, the error based on the $H_2$-norm between the original and reduced order model can be calculated. The formula for the error is similar to equation 2.1. However, the matrices $L$, $C$ and $M$ are now replaced by $L_e$, $C_e$ and $M_e$ respectively. Therefore, the error of the reduced matrix compared to the original matrix, given by $\epsilon$, is computed with the following equation:

$$
\epsilon^2 := \text{trace}(M_e^T \int_0^\infty e^{-L_e^T C_e C_e - L_e t} dt M_e).
$$

(2.2)

As described in section 2.1.5, the error must be normalized. To normalize the error, we divide the squared error of the extended matrices by the squared $H_2$-norm of the original model. Therefore, the normalized error $\bar{\epsilon}$ is given by the following formula:

$$
\bar{\epsilon} := \frac{\epsilon^2}{\gamma^2}.
$$

(2.3)

2.3.1 Lyapunov method

To solve the integral, which is present in the error equation we use the Lyapunov method. The Lyapunov method is explained in Harry L. Trentelman’s book [Trentelman, Stoorvogel, and Hautus, 2002]. The book states the following:

$$
A^T X + XA + Q = 0
$$

$$
X = \int_0^\infty e^{A^T t} Q e^{At}. 
$$

Therefore, to solve the integral of equation 2.1 we let $A = -L$ and $Q = C^T C$ and to solve the integral of equation 2.2 we let $A = -L_e$ and $Q = C_e^T C_e$.

2.3.2 Lyapunov command in MATLAB

To use the Lyapunov command in MATLAB, all eigenvalues of the matrix $A$ should have a negative real part. Nonetheless, the first eigenvalue of the Laplacian matrix is always zero. Therefore, not all eigenvalues of $A$ have a negative real part. In order to prevent this problem, the Laplacian matrices should be adjusted in such a way that the matrix does not have an eigenvalue of zero, while the outcome of the error equation 2.1 should remain unchanged. We introduced the subscript $a$ which stands for adjusted, and subscript $ae$ stand which stands for adjusted matrix of an extended matrix. To prevent the eigenvalue of zero, without changing the solution of the error equation, we adjust the matrices $L$ and $L_e$ as follows:

$$
L_a := L + \mathbf{1} \mathbf{1}^T 
$$

$$
L_{ae} := \begin{bmatrix} L + \mathbf{1} \mathbf{1}^T & 0 \\ 0 & L + \mathbf{1} \mathbf{1}^T P^T P \end{bmatrix}. 
$$

The addition of an all-ones matrix to $L$ creates a matrix that is Hurwitz, which implies that it strictly has positive eigenvalues. Since we let $A = -L$, matrix $A$ strictly has negative eigenvalues. Additionally, the all-ones matrix cancels out to both the transposed output matrix $C$ and the powers of $L$. Put differently, $e^{-L_a t} C^T = e^{-L t} C^T$. Therefore, with the adjustment the Lyapunov command can be used in MATLAB and the solution will be the correct error. These conclusions can also be made after the addition of $\mathbf{1} \mathbf{1}^T P^T P$ to $L$ and the all-ones matrix to $L$ in the adjusted extended Laplacian matrix. Consequently, the adjusted error equations are computed as follows:

$$
\gamma_a^2 := \text{trace}(M^T \int_0^\infty e^{-L_a^T C_a C_a - L_a t} dt M) 
$$

(2.4)

$$
\epsilon_a^2 := \text{trace}(M_e^T \int_0^\infty e^{-L_{ae}^T C_{ae} C_{ae} - L_{ae} t} dt M_e). 
$$

(2.5)
Since the adjustments applied to the Laplacian matrices do not change the solution, equation 2.1 is equal to equation 2.4, and equation 2.2 is equal to equation 2.5. Hence, $\gamma^2 = \gamma'^2$ and $\varepsilon^2 = \varepsilon'^2$. Recalling equation 2.3, it can be concluded that the normalized error is calculated by

$$\bar{\varepsilon} = \frac{\varepsilon^2}{\gamma^2}.$$

### 2.4 MATLAB calculations

Most of the calculations are executed in the main MATLAB program, which is from now on called 'clustering program'. However, some matrices have to be determined beforehand as inputs for the MATLAB program.

#### 2.4.1 Input matrices

The input matrices are the incidence matrix, the weight matrix and the follower-leader matrix. The way in which these matrices are constructed is given in Appendix A. As explained in chapter 2, we focus on random graphs as well. A program is constructed which computes random graphs. This program is given in Appendix B. Lastly, all the different possible partition matrices $P$, corresponding to a certain number of clusters and nodes, are computed by the program in Appendix C. The program returns a list of all the different possible partition matrices, denoted by $P\{n,k\}$. In this term, $n$ is equal to the number of nodes, $k$ is equal to the number of clusters.

#### 2.4.2 Clustering program

The clustering program that computes the optimal partition for a given graph and number of clusters is displayed in Appendix D. When a controllable matrix pair is provided, the program will execute all the calculation as described in chapter 2. The program executes the calculations for all the possible number of clusters, except for the cases where the number of clusters equals one or the number of clusters equals the number of nodes. For each number of clusters, the error is calculated for every possible partition as saved in the list $P\{n,k\}$. For each number of clusters, the smallest error is determined and displayed. Furthermore, the partition that corresponds to the smallest error of each number of clusters is displayed. These partitions are the optimal partitions for that number of clusters. Besides the optimal partitions, the program displays an error plot and a plot of the graph. The combination of the visualization of the graph, the error plot and the optimal partitions table make it possible to analyse the graph. The analyses of the graphs are executed in the next chapter.

#### 2.4.3 Analysed graphs

There are multiple things that can influence the behaviour of a network, such as leaders, structure, weights of edges, among others. To investigate the behaviour of these things we will analyse graphs with different characteristics. First of all, specifically structured graphs such as path graphs, circle graphs and tree graphs are analysed. In these graphs we appoint different nodes as leaders, such that the influence of leaders in combination with the structure of graphs can be analysed. We also analyse random graphs with ten nodes, with either one or three leaders. When analysing a random graph with one leader, the way in which the followers are split up in clusters can be analysed whereas when there are three leaders, the influence of leaders on the clustering can be analysed. Lastly, for some edges the weight is adjusted such that the influence of weights can be investigated. Based on the analysis of the optimal partitions tables and the error plots of all the different graphs we will look for patterns in the clustering of the nodes. These patterns will contribute to the creation of a strategy on how to cluster nodes. We execute the above method to find a strategy for two different cases. For case 1, the output matrix is determined based on the incidence matrix $R$, in which the error is measured considering the weighted differences of the states of neighbouring nodes. For case 2, the output matrix is based on a full matrix, and therefore the error is measured considering the the weighted differences of the states of all other nodes.
Chapter 3

Results

In this chapter, the observations and results of the research are discussed. The two cases are described separately in two sections. The first section describes case 1, in which the output matrix is determined by the incidence matrix $R$. Therefore, the error of case 1 is measured considering the weighted differences of the states of neighbouring nodes. The second section describes case 2, where the output matrix is calculated with the full matrix $\bar{R}$. Hence, for case 2 the error is measured considering the weighted differences of the states of all other nodes. All the analysed graphs along with the corresponding optimal partitions table and error plot are given in Appendix E and F.

3.1 Case 1: Incidence matrix $R$

In this section we analyse multiple different graphs in which the output matrix is based on the incidence matrix. For most graphs we repeatedly assign different nodes as leaders. The following types of graphs are analysed:

- Structured graphs
  - Path graphs
  - Circle graphs
  - Tree graphs
- Random graphs
- Graphs with a largely weighted edge (denoted by ‘weighted graph’)

The results of the analyses are given in Appendix E.

A first clear pattern is already visible by studying the error plots of all graphs: for every graph, as the number of clusters $k$ increases, the error decreases. Furthermore, the error is always between 0 and 1. Based on the theory, an error of 1 is high, while the closer the error is to 0, the better the partition is.

3.1.1 Structured graphs

With the analysis of the structured graphs, given in Appendix E.1, some clear patterns in the clustering are visible. First of all, if the number of clusters is bigger than the number of leaders, each leader is located in an isolated cluster; a cluster containing only one node. The rest of the nodes, all followers, are spread across the other cluster(s).

The error plots also imply the importance of assigning leaders to an isolated cluster. Every time a leader is assigned to an isolated cluster, the error decreases significantly. When all the leaders are in an isolated cluster and the number of clusters increases, the error decreases significantly less.

When all leaders are assigned to an isolated cluster, the followers are gradually spread across the clusters as the number of clusters increases. There are two things which can be observed from the results. Firstly, by looking at the optimal partitions table of for example the path with leader 1 (Figure E.1), the tree with leaders 1 and 7 (Figure E.7) or the circle with leaders 1 and 2 (Figure E.10), the further away nodes are from the leaders, the likelier they are to be in a big cluster. Moreover, when nodes are close to a leader, they are either assigned to a cluster with other nodes
that have the same distance to the leader, or assigned to an isolated cluster. This last pattern is clearly visible in the path graph with leader 5 (Figure E.3) and the circle with leaders 1 and 2 (Figure E.10). Furthermore, if the number of clusters is close to the number of nodes, the nodes far away from leaders are grouped together in a cluster.

3.1.2 Random graphs

The random graphs that are investigated are given in Appendix E.2. Graphs with either one or three leaders are analysed. For the case of one leader, the way in which the followers are split up in clusters can be analysed whereas for the case of three leaders, the influence of leaders on the clustering can be analysed. Since the found patterns are the same for both cases, they are not discussed in separate sections. The observations for the random graphs are similar to that of structured graphs. When there are enough clusters to assign each leader to an isolated cluster, each leader is in an isolated cluster. However, if the number of clusters is not bigger than the number of leaders, the leaders are rarely placed together in a cluster but some leaders are placed together with the followers. In the case where there is only one leader, the leader is separated from the followers directly with two clusters. When we analyse the error plots, again there is a significant change in the slope once every leader is in an isolated cluster. Moreover, when the number of clusters increases, the closer a node is to the leaders the likelier it is to be either assigned to an isolated cluster, or to a cluster with nodes that have similar distances to one or more leaders. Additionally, the closer a node is to a leader, the smaller its cluster is. All these observations are best visible in random graph 6 (Figure E.20) and random graph 10 (Figure E.24).

3.1.3 Weighted graphs

In this section, larger weights are attached to different edges. The edge to which we assigned a larger weight is either attached to a leader and a node on the border of the graph, attached to a leader and the rest of the graph, attached to two followers on the border of the graph, or in the middle of the graph. The magnitude of the weight is chosen in such a way that it is as large as possible, without becoming an uncontrollable graph.

The overall results of each weighted graph are similar to the unweighted graphs. However, some small changes did occur as a consequence of the large weights. First of all, if a leader is connected with a largely weighted edge to a follower, that follower is assigned to an isolated cluster once all leaders have their own isolated cluster. Secondly, if two followers are connected by a largely weighted edge, they will never be in separate clusters. These observations are visible in the weighted graphs given in Appendix E.3.

3.1.4 Conclusions for case 1

To summarize this section, the more clusters there are, the lower the error. The biggest error reductions occur when a leader is taken out of a cluster with followers, and placed in an isolated cluster. Furthermore, the distance of followers to leaders is important for the judgment of which nodes should be together, and which nodes should be in an isolated cluster. If the number of clusters is small, nodes with the same distance to leaders are placed in a cluster together. Moreover, if the number of clusters is approaching the number of nodes, the followers close to a leader are assigned to an isolated cluster, and the followers far away are put in a cluster together. The further away from leaders, the more likely a follower is to be in a cluster with other nodes. Lastly, when larger weights are added, a similar allocation of nodes in the clusters happens. However, some small differences occur. If a follower is connected to a leader with a largely weighted edge, that specific follower is assigned an isolated cluster directly after each leader has an own isolated cluster. Furthermore, if a follower is connected to another follower with a largely weighted edge, they will never be in separate clusters.
3.1.5 Strategy for case 1

Based on the conclusions of case 1, a strategy can be constructed for the clustering of nodes. With a low number of clusters the leaders should be separated from the followers. Each leader must be placed in an isolated cluster and the followers must be placed in one big cluster. For a medium number of clusters, the followers with the same distance to a leader must be placed in a cluster together. If the number of clusters large, the nodes which are close to a leader must be placed in isolated clusters, and the nodes far away from the leaders must be placed in a bigger cluster.

With the addition of larger weights to certain edges, the strategy is almost the same. The only difference is that if a follower is connected to a leader with a largely weighted edge, that specific follower must be assigned to an isolated cluster directly after each leader has an own isolated cluster. Furthermore, if a follower is connected to another follower with a largely weighted edge, they must never be in separate clusters.

3.2 Case 2: full matrix $\overline{R}$

In this section we execute a similar analysis as the previous section. However, contrary to the previous section, in this section we analyse the partitions where the output was based on the full matrix. The results of the analyses are given in Appendix F.

3.2.1 Structured graphs

Firstly, when observing the errors of each graph, it becomes clear that for most graphs the errors for less than 7 clusters are bigger than 1, and therefore the errors are much higher than case 1. The error plots are descending, and the slope are decreasing slightly as the number of clusters increases. Contrary to case 1, the leaders are in clusters together with close-by followers if the number of clusters is low. When the number of clusters increases, two different scenarios can occur. One scenario is that every cluster has a similar size to the other clusters, and each cluster contains nodes that are close to each other. This scenario is clearly visible in the path graph with leaders 1 and 10 (Figure F.2), the tree graph with leaders 3 and 8 (Figure F.9) and the circle with leaders 1 and 4 (Figure F.11). The other scenario is similar to case 1, in which there are many isolated clusters, and only one or two big clusters which contain the nodes that are not in an isolated cluster. This scenario is visible in the path with leader 1 (Figure F.1), the tree with leaders 1 and 7 (Figure F.7) and the circle with leaders 1, 2 and 3 (Figure F.13). In this scenario, the big cluster contains nodes that are far away from the leaders. There are also multiple graphs in which none of these two scenarios occur, such as the path with leaders 1 and 7 (Figure F.4).

3.2.2 Random graphs

An analysis is executed on random graphs with either one or three leaders. For the case of one leader, the way in which the followers are split up in clusters can be analysed whereas for the case of three leaders, the influence of leaders on the clustering can be analysed. The results of the analyses are given in Appendix F.2.

In the case of random graphs with only one leader, there are clear patterns that hold for every analysed random graph. Firstly, the leader is always located in an isolated cluster. Secondly, followers with equal distances to the leader are assigned to a cluster together. Thirdly, when the number of clusters is increasing, the nodes in the cluster that are nearest to the leader are split up, resulting in an isolated cluster for every node close to a leader. Eventually, when the number of clusters approaches the number of nodes, the cluster farthest away from the leader is split up into two isolated clusters.

With the analysis of random graphs based on the full matrix with three leaders, there are no clear patterns that always occur. For some graphs, such as random graph 6 (Figure F.20) and random graph 10 (Figure F.24), each leader is in an isolated cluster when the number of clusters is bigger than the number of leaders. However, for some other graphs this does not hold. An example for this is random matrix 8 (Figure F.22). Furthermore, for some graphs, the followers with equal distance to a leader are placed together in a cluster, for example in random graph 6 (Figure F.20) and random graph 9 (Figure F.23). However, for some other graphs, the followers that are close to each other and close to a leader are assigned to the same cluster. An example
of this is the partitioning of random graph 5 (Figure F.19). Furthermore, the error for each graph with 4 or less clusters is above 1, and therefore the error is much higher than in case 1. Lastly, for most graphs, the error plot has a steep slope until every leader is in an isolated cluster. After each leader is assigned to an isolated cluster, the slope is significantly flatter. An example of this is the error plot of random graph 9 (Figure F.23) and random graph 10 (Figure F.24). However, this does not hold for all graphs, for instance random graph 5 (Figure F.19) and random graph 7 (Figure F.21).

### 3.2.3 Weighted graphs

The example graphs, error plots and optimal partitions table for this subsection are given in Appendix F.3. Similar to in the previous section, the larger weights are assigned to certain edges such that the behaviour of the weights can be analysed. Overall, the partitions are similar to the partitions of the unweighted graphs. However, there are some details which are different. When the largely weighted edge is between a follower and another follower, the two followers will never be in different clusters. Furthermore, if the largely weighted edge connects a follower to a leader, they are in a cluster together when there are six or less clusters. When there are seven or eight clusters, the two nodes are each placed in an isolated cluster.

### 3.2.4 Conclusions for case 2

It is difficult to draw a solid conclusion when the clustering is executed with an output based on the full matrix. The error is much higher than in case 1, and therefore the found partitions are not a good approximation of the original system. The higher error is probably caused by the many connecting edges which are present in the full matrix. Every edge influences the behaviour of the network, and therefore the more edges that are present in a system, the harder it is to predict the behaviour by means of a reduced order model.

There is not one clear pattern in the way of clustering. For some graphs, leaders are assigned to an isolated cluster with a small number of cluster, implying that the leaders must be separated from the followers. However, for other graphs the leaders are in clusters together with followers even for a large number of cluster, implying that it does not matter that a leader is in a cluster with followers. Moreover, for some graphs, the followers that are close to each other are in a cluster together, while for other graphs the followers with equal distance to a leader are assigned to the same cluster. When a larger weight is added, some small differences occur in the partitioning. Firstly, if a follower is connected to a leader by a largely weighted edge, they will be in a cluster together even when the number of clusters is large. Moreover, if a follower is connected to another follower with a largely weighted edge, those two followers will never be in separate clusters. Based on the conclusion for case 2 it becomes clear that there are not enough patterns to base a strategy on. Therefore, further analysis must be executed for case 2 such that a strategy can be developed.
Conclusion

Analysing networks with the size of a power grid is difficult and therefore the focus of the project was on the analysis of networks with ten nodes. The proposed strategy for the ten node networks might however also be used for the analysis of power grids. To verify this, additional research must be executed.

During the research we attempted to find strategies for the clustering based on the cluster-based model reduction technique for two different cases. In case 1 the error was calculated considering the weighted differences of the states of neighbouring nodes, while in case 2 the error was measured considering the weighted differences of the states of all other nodes. By evaluating multiple different graphs, such as structured graphs, random graphs and graphs with a largely weighted edge, we attempted to find a strategy.

For both cases, the error decreased as the number of clusters increased. Therefore, the more clusters a reduced network has, the better it approximates the behaviour of the network. For case 1, a strategy is proposed and can be found in section 3.1.5. For case 2, no clear patterns emerged and therefore no strategy could be proposed. To find a strategy for this case, further research must be executed.
Appendices
Appendix A

Manually created $R$, $M$ and $W$ matrix

\begin{verbatim}
R = [-1  0  0  0  0  0  0  0  0  0;
     1  1  0  0  0  0  0  0  0  0;
     0 -1  1  0  0 -1  0  0  0  0;
     0  0 -1  1  0  0  0  0  0  0;
     0  0  0 -1 -1  0  0  0  0  0;
     0  0  0  0  1  0  0 -1  1  0;
     0  0  0  0  0  0  0  0 -1  1;
     0  0  0  0  0  0  0  0  0 -1;
     0  0  0  0  0  0  1  1  0  0;
     0  0  0  0  0  1 -1  0  0  0]
M = [ 0  0  1  0  0  0  0  0  0  0;
     0  0  0  0  0  0  0  1  0  0]
W = diag([3 1 2 1 2 4 3 2 3 1 1])
\end{verbatim}
Appendix B

Random $R$ matrix generator

```
n=10; %the number of vertices
A = rand(n); %a random matrix of n by n with values ranging between 0 and 1 is constructed
A = (A'+A)/2; %to obtain a symmetric matrix, this step is executed
A = A - diag(diag(A)); %to change the diagonal values are removed, such that they are zero
A = round(0.75*A); %to create a not too dense matrix, the matrix is multiplied by 0.75 and then rounded to zeros and ones
A = graph(A); %creating a graph using an edge list
R = incidence(A); %calculate the graph incidence matrix
R = full(R); %transform it into a full matrix
```
Appendix C

\textit{P} matrix generator

\begin{verbatim}
1 clear P
2 N = 9; % N = number of clusters - 1 (so N=9 gives 10 clusters)
3 P{1,1} = {[1]};
4 for n=1:N
    for k=1:n
        for i=1:length(P{n,k})
            p_ = P{n,k}{i};
            s_ = [p_, zeros(n,1); zeros(1,k), 1];
            if size(P,1) < n+1 || size(P,2) < k+1
                P{n+1,k+1} = {s_};
            else
                P{n+1,k+1}{end+1} = s_;
            end
        end
        for j=1:k
            r_ = zeros(1,k);
            r_(1,j) = 1;
            s_ = [p_; r_];
            if size(P,1) < n+1 || size(P,2) < k
                P{n+1,k} = {s_};
            else
                P{n+1,k}{end+1} = s_;
            end
        end
    end
end
\end{verbatim}
Appendix D

Clustering program

```matlab
[n,~] = size(R); % number of vertices
L = R * W * R'; % Laplacian matrix

clear errors minimalP output
minimalP = cell(1,n);
output = cell(n-2,n);

if n == rank(ctrlb(-L,M)) % controllability check
disp('The system is controllable')
load P10.mat % loading the P matrices

for k = 2:n-1 % clusters that has to be analysed
H2error=0;
lengthP = length(P{n,k}); % number of clusters
C = W^0.5 * R'; % output matrix

% ---------------- Reduced model ----------------
for i = 1:lengthP
Lhat = inv(P{n,k}{i}' * P{n,k}{i}) * P{n,k}{i}' * L * P{n,k}{i}';
Chat = W^0.5 * (P{n,k}{i}' * R)';
Mhat = inv(P{n,k}{i}' * P{n,k}{i}) * P{n,k}{i}' * M;
Rhat = P{n,k}{i}' * R;

% ---------------- Combined matrices ----------------
zerosL = zeros(n,k); % zero matrix created for Lcombined
Lcombined = [L zerosL; zerosL' Lhat]; % Lcombined created
Mcombined = [M; Mhat]; % Mcombined created
Ccombined = [C -Chat]; % Ccombined created

% lyapnov
onesMatrix = [ones(n,n) zeros(n,k); zeros(k,n) ones(k,k)*P{n,k}{i}'];
Ladjustedoriginal = L + ones(n,n);
Ladj ustedcombined = Lcombined+onesMatrix;
Q1 = C'*C;
Q2 = Ccombined' * Ccombined;

Xoriginal = lyap(-Ladj ustedoriginal,Q1);
Xcombined = lyap(-Ladj ustedcombined,Q2);

H2errorCombinedSquared = trace(Mcombined' * Xcombined * Mcombined);
H2errorOriginalSquared = trace(M' * Xoriginal * M);
```

H2error(i) = H2errorCombinedSquared / H2errorOriginalSquared;
end

[minError, minP] = min(H2error);
[maxError, maxP] = max(H2error);

%output for the command window
k
minError;
minimalP[k] = P[n,k]{minP};
output{k-1,1} = minError;
y = ((1:n)’.*minimalP[k])’;
for i = 1:k
    output{k-1,i+1} = mat2str(setdiff(Y(i,:),[0]));
end

errors(k) = minError %errors in a vector
end

figure(1)
m=M’*(1:n)’;
L = R * W * R’;
G = diag(diag(L))–L; %adjacency matrix
H=plot(graph(G));
highlight(H,[m],’NodeColor’,’r’)
title(’Graph’)
errors(:,1) = [];
figure(2);
k1 = [2:k];
plot(k1,errors)
xlabel(’k’)
ylabel(’error’)
title(’Error plot’)
else
disp(’The system is uncontrollable. Therefore, the calculations are not executed.’)
end
Appendix E

Results case 1: incidence matrix $R$

E.1 Structured graphs of case 1

For every figure the layout is the same. The upper left picture displays the graph. The upper right picture displays the error plot. In the table, each row represents an optimal partition: the first row denotes the optimal partition for two clusters, up to and including the last row which denotes the optimal partition for nine clusters. Each column represents a cluster with the indices representing nodes. The first column is not a cluster, but represents the error of the partition. The red nodes are leaders and the blue nodes are followers.

E.1.1 Path graphs

![Path graph with leader 1](image.png)
Figure E.2: Path graph with leaders 1 & 10

Figure E.3: Path graph with leader 5
**Figure E.4:** Path graph with leaders 1 & 7

**Figure E.5:** Path graph with leaders 2 & 8
E.1.2 Tree graphs

**Figure E.6:** Tree graph with leaders 1, 7 & 10

**Figure E.7:** Tree graph with leaders 1 & 7
Figure E.8: Tree graph with leaders 1, 4 & 7

Figure E.9: Tree graph with leaders 3 & 8
E.1.3 Circle graphs

**Figure E.10:** Circle graph with leaders 1 & 2

**Figure E.11:** Circle graph with leaders 1 & 4
FIGURE E.12: Circle graph with leaders 1, 4 & 7

FIGURE E.13: Circle graph with leaders 1, 2 & 3
FIGURE E.14: Circle graph with leaders 1, 2, 3 & 7
E.2 Random graphs of case 1

For every figure the layout is the same. The upper left picture displays the graph. The upper right picture displays the error plot. In the table, each row represents a optimal partition; the first row denotes the optimal partition for two clusters, up to and including the last row which denotes the optimal partition for nine clusters. Each column represents a cluster with the indices representing nodes. The first column is not a cluster, but represents the error of the partition. The red nodes are leaders and the blue nodes are followers.

E.2.1 Random graphs with 1 leader

**Figure E.15: Random graph 1**

**Figure E.16: Random graph 2**
**Figure E.17**: Random graph 3

**Figure E.18**: Random graph 4
E.2.2 Random graphs with 3 leaders

**Figure E.19:** Random graph 5

**Figure E.20:** Random graph 6
Figure E.21: Random graph 7

Figure E.22: Random graph 8
**Figure E.23: Random graph 9**

![Graph 9 Diagram](image1)

<table>
<thead>
<tr>
<th>k</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4094</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>0.2101</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>0.0476</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>0.0216</td>
<td>1</td>
<td>2</td>
<td></td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>0.0071</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>0.0016</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

**Figure E.24: Random graph 10**

![Graph 10 Diagram](image2)

<table>
<thead>
<tr>
<th>k</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5844</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>0.3704</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>0.0780</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>0.0413</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>0.0381</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>0.0177</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>0.0076</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>0.0039</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>
E.3  Weighted graphs of case 1

For every figure the layout is the same. The upper left picture displays the graph. The upper right picture displays the error plot. In the table, each row represents a optimal partition; the first row denotes the optimal partition for two clusters, up to and including the last row which denotes the optimal partition for nine clusters. Each column represents a cluster with the indices representing nodes. The first column is not a cluster, but represents the error of the partition. The red nodes are leaders and the blue nodes are followers.

E.3.1  Weighted graph 1
E.3.2 Weighted graph 2

**FIGURE E.29: Weighted graph 2a**

**FIGURE E.30: Weighted graph 2b**
FIGURE E.31: Weighted graph 2c

FIGURE E.32: Weighted graph 2d
Appendix F

Results case 2: full matrix $\bar{R}$

F.1 Structured graphs of case 2

For every figure the layout is the same. The upper left picture displays the graph. The upper right picture displays the error plot. In the table, each row represents an optimal partition: the first row denotes the optimal partition for two clusters, up to and including the last row which denotes the optimal partition for nine clusters. Each column represents a cluster with the indices representing nodes. The first column is not a cluster, but represents the error of the partition. The red nodes are leaders and the blue nodes are followers.

F.1.1 Path graphs

![Path graph with leader 1](image-url)
FIGURE F.2: Path graph with leaders 1 & 10

FIGURE F.3: Path graph with leader 5
**FIGURE F.4:** Path graph with leaders 1 & 7

![Path graph with leaders 1 & 7](image)

**FIGURE F.5:** Path graph with leaders 2 & 8

![Path graph with leaders 2 & 8](image)
F.1.2 Tree graphs

**Figure F.6:** Tree graph with leaders 1, 7 & 10

**Figure F.7:** Tree graph with leaders 1 & 7
**Figure F.8:** Tree graph with leaders 1, 4 & 7

**Figure F.9:** Tree graph with leaders 3 & 8
F.1.3 Circle graphs

**Figure F.10:** Circle graph with leaders 1 & 2

**Figure F.11:** Circle graph with leaders 1 & 4
Figure F.12: Circle graph with leaders 1, 4 & 7

Figure F.13: Circle graph with leaders 1, 2 & 3
FIGURE F.14: Circle graph with leaders 1, 2, 3 & 7
F.2 Random graphs of case 2

For every figure the layout is the same. The upper left picture displays the graph. The upper right picture displays the error plot. In the table, each row represents a optimal partition: the first row denotes the optimal partition for two clusters, up to and including the last row which denotes the optimal partition for nine clusters. Each column represents a cluster with the indices representing nodes. The first column is not a cluster, but represents the error of the partition. The red nodes are leaders and the blue nodes are followers.

F.2.1 Random graphs with 1 leader

**Figure F.15: Random graph 1**

![Figure F.15: Random graph 1](image1)

**Figure F.16: Random graph 2**

![Figure F.16: Random graph 2](image2)
**Figure F.17:** Random graph 3

**Figure F.18:** Random graph 4
F.2.2 Random graphs with 3 leaders

**Figure F.19: Random graph 5**

![Graph 5](image)

**Figure F.20: Random graph 6**

![Graph 6](image)
**Figure F.21:** Random graph 7

**Figure F.22:** Random graph 8
FIGURE F.23: Random graph 9

FIGURE F.24: Random graph 10
F.3 Weighted graphs of case 2

For every figure the layout is the same. The upper left picture displays the graph. The upper right picture displays the error plot. In the table, each row represents an optimal partition: the first row denotes the optimal partition for two clusters, up to and including the last row which denotes the optimal partition for nine clusters. Each column represents a cluster with the indices representing nodes. The first column is not a cluster, but represents the error of the partition. The red nodes are leaders and the blue nodes are followers.

F.3.1 Weighted graph 1
Figure F.27: Weighted graph 1b

Figure F.28: Weighted graph 1c
F.3.2 Weighted graph 2

**FIGURE F.29: Weighted graph 2a**

**FIGURE F.30: Weighted graph 2b**
FIGURE F.31: Weighted graph 2c

FIGURE F.32: Weighted graph 2d
Bibliography


