Graph Partitioning Algorithms for Control of AC Transmission Networks

Generator Slow Coherency, Intentional Controlled Islanding, and Secondary Voltage Control

Tyuryukanov, Ilya

DOI
10.4233/uuid:a2c1a54a-ab89-4a6a-b9f9-c63241d2c4b8

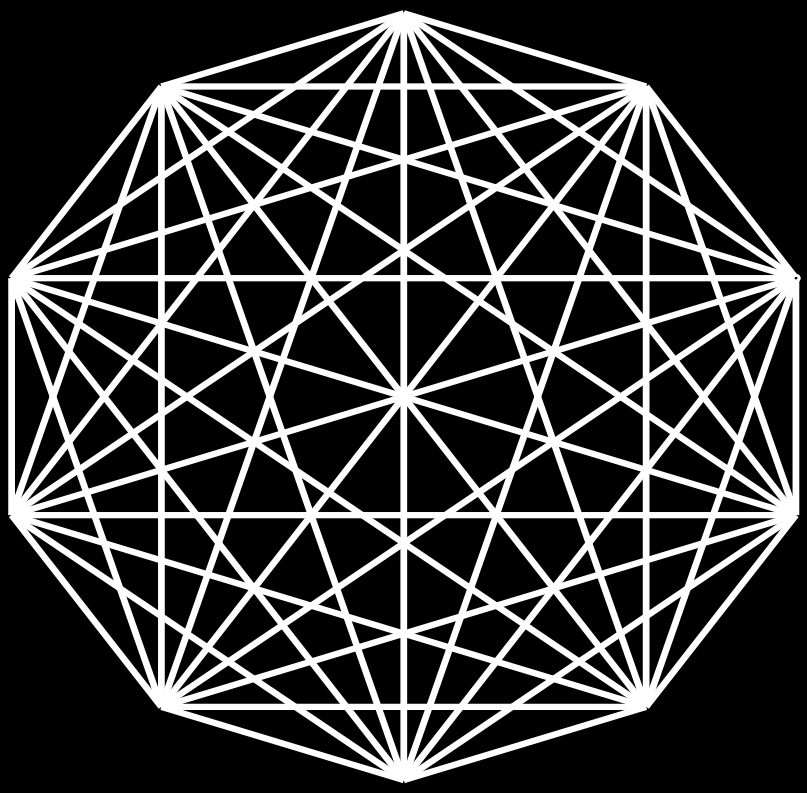
Publication date
2020

Document Version
Final published version

Citation (APA)
Tyuryukanov, I. (2020). *Graph Partitioning Algorithms for Control of AC Transmission Networks: Generator Slow Coherency, Intentional Controlled Islanding, and Secondary Voltage Control*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:a2c1a54a-ab89-4a6a-b9f9-c63241d2c4b8

Graph Partitioning Algorithms for Control of AC Transmission Networks

Generator Slow Coherency, Intentional Controlled Islanding, and Secondary Voltage Control



Ilya Tyuryukanov

Graph Partitioning Algorithms for Control of AC Transmission Grids

Generator Slow Coherency, Intentional Controlled Islanding, and Secondary Voltage Control

Graph Partitioning Algorithms for Control of AC Transmission Grids

Generator Slow Coherency, Intentional Controlled Islanding, and Secondary Voltage Control

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology,
by the authority of the Rector Magnificus Prof. Dr. ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates,
to be defended publicly on
Tuesday 31 March 2020 at 10:00 a.m.

by

Ilya Tyuryukanov

Master of Science in Electrical Engineering and Information Technology,
RWTH Aachen University, Germany,
born in Balashikha, Russia

This dissertation has been approved by

promotor: Dr. Dipl.-Ing. M. Popov
promotor: Prof. ir. M.A.M.M. van der Meijden

Composition of the doctoral committee:

Rector Magnificus,	Chairperson
Dr. Dipl.-Ing. M. Popov	Delft University of Technology, promotor
Prof. ir. M.A.M.M. van der Meijden	Delft University of Technology, promotor

Independent members:

Prof. Dr. P. Palensky	Delft University of Technology
Prof. Dr. H. X. Lin	Delft University of Technology/Leiden University
Prof. Dr. V. Terzija	The University of Manchester
Prof. Dr. X. Kang	Xi'an Jiaotong University
Ir. J. Bos	TenneT TSO B.V.
Prof. Dr. C. Vuik	Delft University of Technology, reserve member

This research was financially supported by the Dutch Research Council (NWO) within the program of Uncertainty Reduction of Smart Energy Systems (URSES).

ISBN 978-94-6384-116-0

Keywords: Dynamic model reduction, generator aggregation, intentional controlled islanding, number of clusters, power network partitioning, secondary voltage control, slow coherency

Printed in The Netherlands by Ipskamp Printing, Enschede

To the memory of my beloved mother.

CONTENTS

SUMMARY

Interconnected electric power systems constitute an important cornerstone of modern developed societies. The economic efficiency of high power transfers from energy rich locations to major load hubs comes at a price of increased power network complexity, higher operational uncertainty, and more complex power system dynamics. The ongoing trends of energy liberalization, growing energy demand, and bulk grid integration of renewable generation additionally increase the complexity and uncertainty levels of modern and future power systems. Enhanced monitoring combined with more adaptive control and protection are often seen as the key components to cope with the described challenges and to enable the transition to a sustainable power system of the future. The advent of synchronized measurement technology (SMT) based on phasor measurement units (PMUs) has stimulated the development of control and protection strategies operating on a set of geographically dispersed power system elements in a coordinated manner. However, the vast size of a modern interconnected power grid precludes controlling and operating it as a single object. Subdividing a power grid into a number of internally coherent control areas is often seen as a means to cope with its inherent complexity and to enable more efficient adaptive control structures.

This thesis focuses on discovering the power system structure to facilitate the definition of control areas for wide-area monitoring, protection and control (WAMPAC) applications. Graph partitioning is seen as a well-developed discipline whose potential is not fully recognized in the power system domain. The research starts by critically reviewing the existing graph partitioning algorithms for their suitability for area identification in power networks. Several auxiliary algorithms are proposed to fix the identified deficiencies of standard graph partitioning approaches and to improve their outcomes. Next, a framework is proposed to choose the number and structure of control areas by modeling a power system as a suitable similarity graph. This research direction concludes in devising a new zoning algorithm for secondary voltage control (SVC) and a new grouping algorithm for generator slow coherency. To confirm the slow coherency findings, some improvements to nonlinear power system model reduction are proposed. Another research direction consists in partitioning power networks with respect to node grouping constraints. These constraints arise in some wide-area monitoring, protection and control (WAMPAC) applications, with generator coherency grouping constraints for intentional controlled islanding (ICI) being a notable example. A new constrained graph partitioning algorithm is proposed that aims to minimize the number of unsatisfied node grouping constraints and favorably compares with the state-of-the-art alternatives. This algorithm is further used as an initialization heuristic for an intentional controlled islanding (ICI) approach based on mixed-integer linear programming (MILP). Besides the applications in WAMPAC that motivated this thesis, the developed contributions could be useful for other power system grouping problems, including reduction or large datasets and decomposition of large optimization problems.

SAMENVATTING

Onderling verbonden elektrische energiesystemen vormen een belangrijke hoeksteen van moderne ontwikkelde samenlevingen. Voor de economische efficiëntie van de overdracht van hoog vermogen van energierijke locaties naar belangrijke belastingsknooppunten dient een prijs betaald te worden in de vorm van verhoogde complexiteit van het energienetwerk, grotere operationele onzekerheid en complexere dynamiek van het energiesysteem. De voortgaande tendens van energieliberalisering, groeiende vraag naar energie en de integratie van hernieuwbare energie-opwekking in het bulknetwerk verhogen bovendien de complexiteit en niveaus van onzekerheid van moderne en toekomstige energiesystemen. Verbeterde monitoring in combinatie met adaptieve controle en beveiliging worden vaak gezien als de sleutelcomponenten om de beschreven uitdagingen aan te gaan en de transitie naar een duurzaam energiesysteem van de toekomst mogelijk te maken. De opkomst van gesynchroniseerde meettechnologie (synchronized measurement technology, SMT) op basis van fasevector-meeteenheden (phasor measurement units, PMU's) heeft de ontwikkeling van controle- en beveiligingsstrategieën gestimuleerd die werken op basis van een stel geografisch verspreide elementen van energiesystemen op gecoördineerde wijze. De enorme omvang van een modern, onderling verbonden stroomnetwerk sluit echter uit dat dit als een enkel object kan worden bestuurd en bediend. Het onderverdelen van een stroomnetwerk in een aantal intern coherente besturingsgebieden wordt vaak gezien als een middel om om te kunnen gaan met de inherente complexiteit ervan en om efficiëntere adaptieve besturingsstructuren mogelijk te maken.

Dit proefschrift concentreert zich op het ontdekken van de structuur van het energiesysteem om de vaststelling te vergemakkelijken van besturingsgebieden voor breedgebied toezicht, beveiliging en regeling (wide-area measurement, protection, and control, WAMPAC)-toepassingen. De partitie van grafen wordt gezien als een goed ontwikkeld vakgebied waarvan het potentieel niet volledig wordt onderkend binnen het domein van energiesystemen. Het onderzoek begint met het kritisch beoordelen van de bestaande algoritmen voor partitie van grafen op hun geschiktheid voor identificatie van een deelgebied in energienetwerken. Er worden verschillende hulpalgoritmen voorgesteld om de geïdentificeerde tekortkomingen van standaardbenaderingen voor partities van grafen te herstellen en de resultaten daarvan te verbeteren. Vervolgens wordt een raamwerk voorgesteld om het aantal en de structuur van besturingsgebieden te kiezen door een energiesysteem te modelleren als een geschikt similariteitsgraaf. Deze onderzoeksrichting eindigt met het ontwikkelen van een nieuw indelingsalgoritme voor secundaire spanningsregeling (secondary voltage control, SVC) en een nieuw groeperingsalgoritme voor trage coherentie van generatoren. Om de bevindingen van de trage coherentie te bevestigen, worden enkele verbeteringen voor de reductie van het model van niet-lineaire energiesystemen voorgesteld. Een andere onderzoeksrichting bestaat uit het partitioneren van energienetwerken met betrekking tot beperkingen in de

groepering van knooppunten. Deze beperkingen doen zich voor in sommige WAMPAC-toepassingen, waarbij beperkingen voor de groepering van generatorcoherentie voor gecontroleerde netwerkscheiding een opmerkelijk voorbeeld zijn. Er wordt een nieuw algoritme voor beperkte partitie van grafen voorgesteld dat tot doel heeft het aantal niet tot tevredenheid stemmende beperkingen voor de groepering van knooppunten te minimaliseren, een algoritme dat gunstig afsteekt bij de meer geavanceerde alternatieven. Dit algoritme wordt verder gebruikt als een initialisatie-heuristiek voor gecontroleerde netwerkscheiding op basis van gemengd geheeltallige lineaire programmering. Naast de toepassingen in WAMPAC die de motivatie vormden voor deze thesis, kunnen de ontwikkelde bijdragen nuttig zijn voor andere problemen met de groepering van energiesystemen, waaronder reductie of grote datasets en ontleding van grote optimalisatieproblemen.

1

INTRODUCTION

1.1. BACKGROUND AND MOTIVATION

The electric power system is often cited as the largest and most complex machine ever devised by man [1]. Although this statement is very impressive, the present level of development was achieved through overcoming many engineering and socioeconomic challenges that were pervasive during all of history of the electric power industry. This was the process that allowed the power system to progress from the early power plants supplying local customers to a sophisticated network of many thousands of nodes linked by high-voltage transmission lines and spanning whole continents.

Despite the achieved high security and efficiency, the modern-day power systems are undergoing another massive transformation that is largely dictated by the demand for sustainable energy supply. It is envisioned that the increasing complexity and faster power system dynamics arising through this transition will be contained by the new enhanced monitoring combined with the adaptive control and protection [2]. The groundwork for this has been laid by the large progresses in computing, digital signal processing (DSP), information and communications technology (ICT) [3, 4] during the recent decades. Nevertheless, the control and protection methods that could fully take advantage of the current level of ICT still remain an active research field.

Out of many open questions related to the online control of future power grids, this thesis focuses on the idea of power network partitioning into zones, clusters or areas that is common to a large number of existing and prospective applications in power system operations, control and protection. The great potential of partitioning is due to the clustered structure of bulk electric power grids, which are typically composed of smaller sub-grids with varying degrees of mutual interaction. And while most of the existing applications define their control areas based on the historic asset ownership, the consideration of the actual state and physical properties of the network is becoming more important and sometimes inevitable as power systems move towards faster dynamics and greater operational uncertainty.

1.1.1. MODERN RISKS TO POWER SYSTEM SECURITY

The two recent challenges that have deeply impacted the well-established operational principles of the late XX century are the deregulation of the electricity industry [5, 6] and the need for massive integration of renewable energy sources into existing power grids [7, 8]. The transition to the market-based operation of electric power systems has led to an increase in cross-border trading, as electricity imports are often cheaper than the domestic generation. Consequently, the inter-area tie lines initially designed for greater frequency stability become increasingly used for heavy power transfers often leading to the lowering of safety margins. A classic example of the dangers associated with the intensive inter-regional energy trade is the Italian blackout of 2003 that was initiated by a cascading tripping of the highly loaded transmission lines between Italy and Switzerland [9]. The cost-reduction objective of the market-based power system operation is also leading to more frequent long-distance power transfers from cheaper energy sources leading to increased grid congestion [10]. In general, the whole paradigm shift of separating the vertically integrated electric companies into generation, transmission and distribution utilities coupled through the market has profound consequences for the whole areas of power system planning, operation and security.

However, the ongoing transition towards sustainable energy may cause even more significant changes to the electric power grids. The most widely available and promising renewable generation technologies are the switch mode converter interfaced photovoltaics (PV) and wind energy (WE) systems. Due to their low or zero inertia, power electronics based generators adversely affect the power system frequency dynamics by increasing the rate of change of frequency (ROCOF) and decreasing the lowest frequency value (NADIR) following a major power imbalance. The weather-dependent power output of PV and WE systems increases the uncertainties in power system planning and operation [11], which may also contribute to the higher likelihood of power system outages, cascading events and large blackouts.

The introduction of a large number of bulk renewable power plants inevitably leads to the demand for new transmission capacity [12], including the building of new high voltage DC (HVDC) and medium voltage DC (MVDC) lines. Building new interconnections is also seen an important measure to cope with the intermittent nature of the renewable energy sources. For example, the European North Seas Countries Offshore Grid Initiative (NSCOGI) proposes an entire new offshore energy grid in the North Sea to transfer the energy from the offshore wind farms and to enable more possibilities for power exchange between the participating countries [13]. In addition, new transmission lines can be built to balance the market energy prices over different areas [14].

Therefore, both the liberalization of the electricity industry and increasing grid penetration of renewables tend to make the system structure more coupled and complex, which often prompts enhancements to system control and protection. For example, the increasing meshing of the French power grid has required the coordination of the secondary voltage control between the neighboring control zones to circumvent the growing zone coupling [15]. And the Northeast blackout of 2003 in the USA has shown that the conventional out-of-step protection relays may become prone to maloperation in highly interconnected power networks [16].

The higher frequency of wide-area system disturbances and blackouts is another indicator associated with the growing complexity and unpredictability of modern and future electric power grids. Indeed, the past two decades have seen an unusually large number of such events. To illustrate this, some notable power outages since 2003 are listed in Table 1.1 [17, 18, 19, 20, 21, 22] (many lesser blackouts are not included).

Date	Location	Population affected
14 Aug 2003	Northeast USA and Eastern Canada	50 million
02 Sep 2003	Southern Peninsular Malaysia	10 million
23 Sep 2003	Southern Sweden and Eastern Denmark	4 million
28 Sep 2003	Italy	55 million
12 Jul 2004	Greece	5 million
25 May 2005	Moscow Region, Russia	5 million
18 Aug 2005	Java and Bali, Indonesia	100 million
24 Sep 2006	Pakistan	140 million
04 Nov 2006	Europe (UCTE power system)	15 million
26 Apr 2007	Colombia	41 million
10 Nov 2009	Brazil and Paraguay	85 million
14 Mar 2010	Chile	15 million
04 Feb 2011	Brazil	40 million
08 Sep 2011	California and Arizona, USA	8 million
24 Sep 2011	Chile	9 million
14 Jan 2012	Marmara Region, Turkey	20 million
30 – 31 Jul 2012	India	620 million
01 Nov 2014	Bangladesh	150 million
26 Jan 2015	Pakistan	140 million
31 Mar 2015	Turkey	70 million

Table 1.1: Notable large-scale power outages

The recent large blackouts often highlighted the existing inadequacies in monitoring, control and protection of the time-evolving power systems. Multiple expert groups that investigated the outages agree on the following technological innovations to improve the power system resiliency and minimize the risks of blackouts [17, 23, 24]:

- Real-time wide-area power grid monitoring and control
- Coordinated emergency controls
- Use and enhancement of system integrity protection schemes (SIPS), special protection systems (SPS), and remedial action schemes (RAS)
- Online dynamic security assessment
- Improved reactive power management
- Adaptive relaying
- Use of flexible AC transmission systems (FACTS) devices and HVDC

This conclusion may serve as a confirmation of the great role of advanced control and protection algorithms in future power systems.

1.1.2. SMT AND WAMPAC

Possibly the most well-recognized technological platform for the real-time power grid monitoring is the phasor measurement unit (PMU) based synchronized measurement technology (SMT) [3, 25]. The PMU devices are able to estimate the phasors of voltage and current sinusoidal waveforms at synchronous time intervals enforced by the clock signal from satellite navigation systems (e.g., GPS). The standard PMU reporting rates are specified in the IEEE Standard C37.118.2-2011 both for 50 Hz and 60 Hz systems, which are given in Table 1.2.

System frequency	50 Hz			60 Hz					
Reporting rates, Hz	10	25	50	10	12	15	20	30	60

Table 1.2: Standard PMU reporting rates

The standard report rates represent a minimum, and higher PMU rates (e.g., twice the system frequency) can be used instead [26], which may be advantageous for many real-time applications. However, even the minimum report rates given in Table 1.2 are much higher than the typical update rates of the conventional supervisory control and data acquisition (SCADA) systems, which normally require about 5–30 seconds for an update of the system state estimate [27].

The PMU measurements can be transmitted to higher-level entities called phasor data concentrators (PDCs) and super PDCs, where the measurements from different grid locations are time-aligned to yield the wide-area snapshots of electrical variables with a high update rate. The PDCs and super PDCs typically host a data storage and a number of applications. Some examples of the known PMU applications include [3, 25, 28]:

- Real-time power system monitoring
- Oscillation detection and monitoring
- Phasor-based state estimation
- Real-time estimation of system model parameters
- System model validation
- Real-time congestion management
- Identification potential malfunction of devices in the grid

The value of a deployed SMT platform is often assessed by the number and quality of control room applications utilizing the data provided by the PMUs [29]. In this regard, a lot of progress has already been achieved in applications related to power system monitoring, static analysis and data management (e.g., PMU-based state estimation, post-event analysis, oscillation monitoring etc), which are commonly understood to belong to the family of wide-area monitoring systems (WAMS) applications. A higher class of SMT-enabled platforms is called wide-area monitoring and control (WAMC)

or WAMPAC systems. Some notable examples of WAMPAC applications have been described in [17]:

- Wide-area controls to maintain voltage profiles and reactive power reserves
- Wide-area oscillation damping control
- Wide-area control of phase shifting transformers and FACTS devices
- Controlled network separation including generation and load shedding to maintain system frequency and stability
- Intelligent load-shedding to maintain voltage profile and system stability

The area of WAMPAC applications is less mature compared to WAMS and largely constitutes an active research topic [17]. According to [29], Bonneville Power Administration, which was among the early adapters of the SMT technology, set the following goal in mid-2000s:

It is time to move forward from wide-area monitoring to wide-area controls.

The most recent reviews of the current state of the art of the WAMS and WAMPAC technologies (e.g., [28]) demonstrate the large progress achieved by some utilities over the past two decades. However, the goal of transitioning from WAMS to WAMPAC still remains very actual, and multiple envisioned WAMPAC applications still require more research effort to be implemented.

1.1.3. AREA-BASED PROTECTION AND CONTROL

Due to the practical infeasibility of controlling a large synchronous AC power grid as one whole, many existing and prospective power system control and protection functions use the concept of zones or areas in some form. According to Cotilla-Sanchez et. al. [30, 31], the existing applications in power system planning and operations that require the definition of zones or areas include operational security analysis, resource adequacy assessment, zonal pricing, zone-based voltage control schemes, automatic generation control (AGC), area control error (ACE) calculations, reserves scheduling, and load deliverability assessment.

With the advent of the WAMS and WAMPAC, the notion of areas may gain even more significance, as the whole concept of WAMPAC implies the coordinated control and protection over network zones or areas. It is also anticipated that the control areas involved in the WAMPAC applications should increasingly consider the physical properties of the power grid in order to address the adaptive and dynamic nature of the WAMPAC control objectives. This is in contrast to the majority of existing applications, which tend to define the control areas based on the historic asset ownership. To provide an overview, several examples of existing and novel applications requiring the definition of areas based on physical principles are summarized below.

Generator slow coherency identification deals with the task of partitioning the power system into groups of weakly coupled generators [32]. The underlying theory is based on the modal decomposition of the linearized electromechanical model of

power system [33, 34]. The small oscillatory eigenvalues of this model correspond to the slow interarea modes of the system, and their corresponding eigenvectors describe the respective rotor angle mode shapes. With this information, it is possible to identify the network areas with a low dynamic coupling and to perform model reduction by using specialized coherency identification and area aggregation algorithms [34, 35].

Online dynamic security assessment (DSA) is concerned with determining the ability of a power system to maintain stability and operational limits without load interruption under a large set of probable contingencies. The online variety of DSA assesses security at regular time intervals (typically once in several minutes) using the most recent information about the power system state obtained from the SCADA [36]. The accurate stability assessment usually requires computationally demanding time-domain simulations, which is in contradiction with the real-time nature of online DSA. The solution often lies in the reduction of the full-scale power system model into the (unreduced) *study area* of interest and an *external equivalent* representing the rest of the system (i.e., the *external system*). To obtain a high-fidelity reduced model, the study area and the external system should be defined in a way that minimizes the dynamic and power flow coupling between them [37]. Moreover, the external equivalent should preserve the dominant low-frequency dynamics of the external system. These requirements motivate the use of the generator slow coherency identification methods in combination with specialized power flow preserving equivalencing techniques in online DSA [38].

Online voltage security assessment (VSA) is a subcategory of online DSA [36] that is additionally characterized by the computation of *critical voltage control areas (VCAs)*, which are also known as *voltage collapse areas*. VCAs represent the subareas of the study area which can be prone to the loss of voltage stability under certain contingencies. The computation of such VCAs is performed by the Q-V modal analysis [1, 36]. After the VCAs are computed, a reactive reserve requirement to prevent the possible instability is computed for each of them.

Area-based PMU placement is largely associated with the idea of monitoring power system dynamics through a reduced set of measurements. The goal is to partition a large-scale power network into a set of areas with coherent dynamics and to select in each area a *medoid bus* as the bus most representative for the dynamics of its area [39]. As the resulting areas should be dynamically coherent, the notion of generator slow coherency plays an important role in this application. To avoid the limitations associated with the classical model-based coherency approaches [32, 34], many authors prefer to cluster generator signals obtained from PMUs to obtain the areas [39, 40]. The selection of monitoring buses based on cohesive network areas may also be promising for a number of wide-area protection approaches based on the online computation of bus vulnerability indices (e.g., [41]).

Intentional controlled islanding (ICI) is an adaptive, corrective measure that aims to limit the spread of disturbances across the grid by separating it into self-sustainable islands [42]. The generator coherency requirement is important

because non-coherent generators may loose synchronism after separation [35, 43]. Thus, generator coherency algorithms are often considered as an important pre-step of ICI. A more complete description of ICI and requirements to identified islands is given in Section 2.2.2.

Parallel power system restoration (PPSR) aims to restore a collapsed power system in parallel, thus accelerating the overall restoration process [44]. This is achieved by separating the blackout-affected area into a number of sections. The sections formed following a blackout should consider the most recent information about the status of generating units, the assignment of generating units into *cranking groups*, the status of lines and circuit breakers, and the predicted load levels [45]. Each cranking group should include at least one *blackstart unit* that should provide the cranking power to the remaining generating units (*non-blackstart units*) in the group. After the defined sections are reenergized in parallel, they are resynchronized to restore the normal network operation.

Zone-based secondary voltage control (SVC) is the second level of hierarchical wide-area voltage control analogous in its purpose to load-frequency control (LFC) within AGC. With zone-based secondary voltage control (SVC), a large power system area (e.g., a national or provincial power grid) is subdivided into a number of voltage control zones (VCZs) featuring cohesive voltage profiles [46, 47]. Each VCZ is controlled by regulating the voltage of specially selected *pilot nodes* to their reference. A more complete description of SVC and its role in hierarchical voltage control is given in Section 2.2.1.

Wide-area voltage protection (V-WAP) can be implemented on top of the control zone structure of SVC [41]. If hierarchical zone-based voltage control is implemented, the closeness of controlling generators of a VCZ to their reactive power limits can be used as a risk indicator for voltage instability [41]. This observation only remains valid if the tertiary voltage control (TVC) is active to lower the SVC voltage setpoints up to an acceptable minimum when the increased system stress makes it difficult to maintain the economically optimal voltage profile [48].

The above applications provide a non-exhaustive set of examples of the usefulness of adaptive area definition in the context of power systems. Some of them will be given more attention in the subsequent sections of this thesis.

1.1.4. AREA IDENTIFICATION IN POWER NETWORKS

As it may become clear from the overview given in the previous section, the types of areas required by various applications can be quite different. For example, online VSA is specifically looking for particular areas that may experience a voltage collapse, and it is not needed to partition the whole power system to estimate these areas. On the contrary, the SVC structure is usually required to cover all high voltage (HV) buses, which is achieved by partitioning of the power system into a number of VCZ. And many applications dealing with electromechanical dynamics (e.g., DSA, ICI) require extra *node grouping constraints* that assign certain coherent generators to a specific area or island

[43]. These node grouping constraints also appear in PPSR, as generator cranking groups should not be split between multiple sections [45].

For the majority of listed applications, some form of clustering method is commonly used to identify the required areas. In the case of model-based generator slow coherency estimation, the slow eigensubspace of the electromechanical system model is clustered with the specialized [33, 34] or general-purpose [49] algorithms. In addition, many clustering methods have been proposed to estimate the generator coherency or coherent network areas from measured signals (e.g., from PMU data) [39, 40, 50]. Zone definition for wide-area voltage control shows a long history of utilization of clustering algorithms, possibly starting from the paper of Lagonotte et. al. on electrical distances [46]. Other well-known approaches include the clustering of voltage sensitivity matrix in [47, 51] and the so-called *Var control space (VCS)* method [52]. If node grouping constraints are present, constrained clustering or constrained graph partitioning algorithms are commonly employed. For example, Ding et. al. used a constrained spectral clustering algorithm to partition power networks for ICI under the consideration of generator coherency constraints [43, 53]. This line of work was extended to be applied to PPSR under the consideration of generator cranking groups constraints in [54]. Some other references utilizing constrained clustering or constrained graph partitioning in the context of ICI and PPSR include [45, 55, 56, 57, 58, 59, 60, 61].

Another important approach for adaptive area identification is mathematical optimization, including mixed-integer programming (MIP) and its subclass mixed-integer linear programming (MILP). The recent big advances in solver technology have made it feasible to solve the exact formulations of multiple discrete optimization problems in power system such as unit commitment (UC) or transmission expansion planning (TEP) [62, 63, 64]. Many area identification problems can be formulated as discrete graph partitioning problems subject to a number of constraints (e.g., minimal area size, area connectedness etc). However, this approach has certain limitations, as the objective function and constraints should be linear to enable feasible solution times on realistically-sized networks (i.e., a MILP formulation is desirable). However, even if a MILP area identification model has been devised, the computation time is likely to grow very fast with the network size due to the NP-complete nature of discrete graph partitioning problems [65, 66]. To confirm this observation, a comparison of the exact MILP-based network partitioning with a basic spectral clustering algorithm can be found in [67].

When the desired optimization objective or its constraints take a complex nonlinear form, metaheuristic optimization techniques may show good results [30]. While this type of optimization is suitable for the most general type of problems, it also lacks multiple features that characterize the success of MILP (e.g., access to the linear relaxation, various strong results in polyhedral theory, optimality gap computation etc). Therefore, it often takes a considerable time for metaheuristics to find an optimum, and the global optimality of the result is not guaranteed. Nevertheless, a well-designed metaheuristic may be more efficient than MILP at finding good, but suboptimal solutions.

1.2. RESEARCH CHALLENGES AND PROBLEM DEFINITION

The present research is motivated by the increasing risks to the electric power infrastructure due to the increased uncertainties and faster dynamics of future power systems. The

main focus is on the clustering-based area identification algorithms for the existing and prospective WAMPAC applications. This seemingly broad scope can be explained by the inherent links between many WAMPAC applications in terms of area definition.

> *In other words, the problem of identifying control areas is at the core of multiple important power system applications, and many of them pose similar requirements to their areas. Therefore, it is promising to emphasize the relevant similarities instead of studying each application in isolation.*

For example, it has been discussed in Sections 1.1.3–1.1.4 that multiple WAMC applications require internally cohesive areas that are well decoupled from each other, and some other WAMPAC applications add node grouping constraints to this requirement.

1.2.1. CLUSTERING TERMINOLOGY

Before delving into the peculiarities of clustering-based area identification in power systems, it is useful to clarify some terminology that has been used intuitively so far.

Cluster denotes a group of objects that are closely related (or similar) among themselves and weakly related (or dissimilar) to other objects that do not belong to the group.

Clustering denotes a general procedure of finding clusters in a set of objects.

Partitioning is a clustering procedure that assigns each object to a single cluster (i.e., partitions the dataset). By the definition, the partitions may not overlap.

Partition (or block) is a cluster obtained as the result of partitioning.

Graph clustering is a clustering procedure defined on a network (also called graph), in which network nodes are the clustered objects.

Graph partitioning is a graph clustering procedure, in which every network node has to be assigned to a single partition (i.e., the partitions may not overlap).

Constrained clustering is a clustering procedure that forms clusters based on the provided similarities between objects and a set of constraints fixing the relationships between certain pairs or groups of objects.

Constrained graph partitioning is defined for this thesis in a narrow sense of partitioning a network with respect to node grouping constraints (additional constraints may or may not be present).

Unconstrained graph partitioning is defined for this thesis as a graph partitioning procedure that does not include node grouping constraints (other constraints such as cluster size constraints may or may not be present).

Connected component is a subset of nodes in a graph that are connected to each other by paths and are not connected to any node outside of the subset.

Connectedness is the property of a network cluster, such as area, zone, or island, to consist of a single connected component.

Area is an internally connected part of an electric power network. Areas are typically well-connected internally and loosely coupled with the rest of the network, which makes them conceptually similar to clusters.

Zone is the same as area, but smaller in size. Large network areas can be partitioned into smaller zones for the purposes of monitoring and control.

Island is an area electrically separated from the rest of the grid or planned to be separated in case of an emergency condition.

The introduced definitions have a large degree of overlap, as it is common across many research disciplines [66, 68]. For example, many algorithms of essentially partitioning nature such as k-means are commonly referenced as clustering algorithms. The exact meaning of constrained graph partitioning varies across different technical domains. For this thesis, the definition of constrained graph partitioning is closest to the typical problem setting of constrained spectral clustering [57, 69], and unconstrained graph partitioning is defined to complement the constrained one. Some of the definitions above are specifically given for *similarity-based clustering* in which a strong relationship between a pair of objects is expressed by a large number. However, analogous definitions can easily be given for the dissimilarity-based clustering, in which relationships between objects are given in terms of distances [70].

Given the above definitions and observations, the terms cluster, partition, and area are used interchangeably with the basic meaning of *partition* whenever this meaning is clear from the context. The same logic applies to the terms graph clustering and graph partitioning, with the term partitioning being exclusively used in relation to networks. Additionally, the terms network and graph are used synonymously.

1.2.2. RESEARCH CHALLENGES

Although the clustering-based area identification is well-established and tends to perform much faster than the optimization-based alternatives (especially on large datasets), it often shows the following drawbacks:

1. Many clustering algorithms fail to identify the correct clusters for certain input datasets due to intrinsic biases caused by their heuristic or approximate nature. In fact, it is well-known that the popular k-means or k-medoids methods tend to return convex-shaped clusters of rather balanced sizes [71], while some variations of agglomerative hierarchical clustering (AHC) are vulnerable to outliers [72].
2. Clustering algorithms are inherently inflexible, as they usually do not allow a significant modification of the underlying objective function and constraints. For example, introducing a minimum cluster size condition proves to be very hard for the vast majority of clustering algorithms, although this becomes a trivial task with mathematical optimization.
3. Clustering algorithms typically aim to approximately solve some important NP-hard problems [66, 73, 74, 75]. Therefore, the clustering outcome is not guaranteed to provide an optimal or even feasible solution to the initial hard problem. On the other hand, clustering algorithms are usually guaranteed to run in a foreseeable short amount of time (i.e., the polynomial time complexity).

The above limitations explain the high degree of specialization among clustering algorithms and thus the absence of a universal clustering algorithm suitable for any type

of application. These limitations may also be at odds with some of the common power system requirements to control areas [30, 43, 76]:

1. Area cohesiveness (high intra-area connectivity)
2. Area separation (low inter-area connectivity)
3. Balanced area sizes (no very small areas)
4. Area connectedness
5. Fulfillment of node grouping constraints

Many popular general-purpose clustering algorithms (e.g., k-means or AHC) tend to perform suboptimally w.r.t. the above requirements. For example, the area cohesiveness and separation criteria often suffer from the intrinsic biases of the mainstream algorithms, which do not well agree with the specific goals of power system area identification. The inexactness of clustering algorithms (the third clustering limitation) also exacerbates this problem. The difficulty of ensuring balanced cluster sizes was mentioned in the discussion of the second clustering limitation.

The area connectedness requires any node to have a connection to any other node in its area going solely through the nodes of that area. The fulfillment of this condition requires to respect the interconnection structure of the power network, which is not included into the majority of standard clustering algorithms, including k-means, AHC, gaussian mixtures, spectral clustering etc [71]. Finally, the fulfillment of node grouping constraints, when combined with the area connectedness requirement, requires from a clustering algorithm not only to find areas as connected components in the power network, but also to include the prescribed nodes into each such connected component. This problem is clearly NP-hard and may have no feasible solutions for some configurations of node grouping constraints, as it can be linked to the *Steiner tree packing problem* [77, 78].

1.2.3. RESEARCH QUESTIONS

The research questions of this thesis aim to tackle the following higher-level objective from different angles:

> *To reconcile the inherent limitations of clustering with the power system requirements to area identification, while introducing novelty when beneficial.*

To achieve this objective, the following research questions are formulated:

I. What are the implications of high area cohesiveness and separation on the efficiency of WAMPAC applications and how can they be assessed?

II. How to achieve a greater control over the clustering results (e.g., to avoid very small areas), while not compromising the computational efficiency?

III. How to ensure area connectivity when applying clustering algorithms to identify areas in power networks?

IV. How to achieve a high degree of satisfaction of node grouping constraints by using constrained clustering algorithms while ensuring area connectivity?

V. How to determine an optimal number of areas for various power system analysis and WAMPAC applications?

VI. How to enhance clustering algorithms to satisfy a larger number of power-system related constraints in a timely manner?

As it can be seen, the research questions revolve around achieving the maximal results by using computationally-efficient clustering algorithms to identify power system areas (in their various forms). This is justified by the online character of many emerging power system applications (e.g., ICI, adaptive SVC) requiring fast solution times that may be unattainable with solely classical optimization methods or metaheuristics.

1.3. RESEARCH APPROACH

From many types of clustering algorithms, this thesis specifically focuses on graph clustering and graph partitioning. The reason for this is twofold: electric power networks can be naturally mapped to graphs, and many seemingly non-graph data (e.g., power flow sensitivities) can be seamlessly represented through graphs.

Among many graph partitioning algorithms, spectral clustering based approaches [74, 79] play the major role in this thesis. This is due to their fundamental mathematical nature and multiple extensions into clustering of labeled data [57, 69, 80]. To realize this conclusion, an extensive literature survey into different graph partitioning methods has been done. An important outcome of the literature survey is the selection of quality metrics that well agree with the general requirements to power system control areas listed in Section 1.2.2.

The considered power system applications can be subdivided according to the presence or absence of node grouping constraints. These constraints are not required for generator slow coherency analysis and SVC, which are the first two applications considered in this thesis. The node grouping constraints are required for ICI in the form of coherent generator groups and for PPSR in the form of generator cranking groups (cf. Section 1.1.3). The cranking groups of PPSR can be handled in a similar fashion to the coherent generator groups of ICI [45, 54], which is the main reason not to consider PPSR in this thesis. Due to many types of power system constraints in ICI, this application is additionally modeled as a MILP, whereby the constrained graph partitioning serves as an initialization heuristic to reduce the solution time. An additional topic is the use of graph pre-processing and post-processing algorithms to improve the clustering results. The pre-processing and post-processing are largely independent from the core method that partitions the graph and can be applied to improve the performance of many clustering methods.

The studied data consists solely of static graph-theoretic models, and no power system transients had to be processed. For this reason, the MATPOWER toolbox running in MATLAB [81, 82] is the de-facto main research tool. MATPOWER contains open-source codes for power flow and optimal power flow (OPF), which are useful to retrieve the relevant static data, as well as a number of test models including large-scale networks. However, MATPOWER alone is not enough for all required tasks, so a number of interconnections have been established to facilitate the data exchange between the specialized

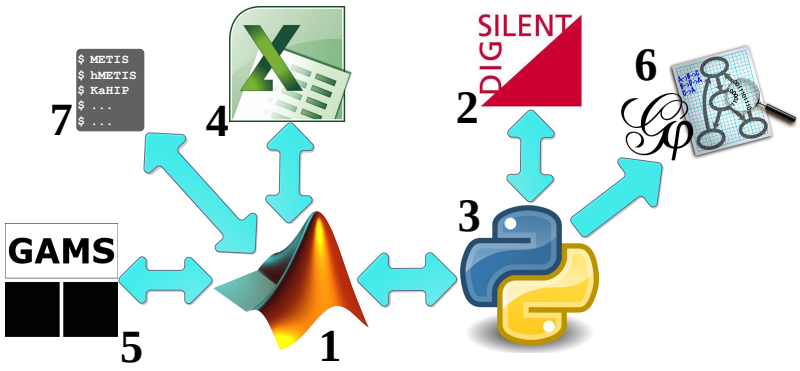


Figure 1.1: Software interconnections and dataflows

software tools. The following list describes the functionality of each software, while Figure 1.1 illustrates the dataflows.

1. MATLAB [82] is the most used software. The MATPOWER toolbox and its uses were described above. The Power System Toolbox (PST) [83] is another external MATPOWER package, which is relevant due to its collection of generator coherency algorithms and related modeling scripts. The efficient matrix computations in MATLAB are the reason to implement the majority of proposed methods in its scripting language. MATLAB also provides built-in interfaces with Python (since MATLAB release 8.4), Java, Microsoft Excel, and the operating system.
2. DIgSILENT PowerFactory [84] is a specialized power system software that has mainly been used for time-domain RMS simulations. It also contains a number of additional power system models and a number of useful functionalities (e.g., contingency analysis). PowerFactory provides a built-in interface with Python.
3. Python [85] is mainly used as an interface environment. Python itself serves as the link between MATLAB and PowerFactory. The Python package igraph is used to connect MATLAB with the graph visualization tools Gephi and GraphViz.
4. Microsoft Excel is a popular spreadsheet software that has mostly been used for importing data (e.g., generator dynamic data tables from [86]).
5. GAMS [87] is an optimization modeling language and software containing links to several state-of-the-art optimization solvers (e.g., IBM® CPLEX®) that is used to model and solve ICI through MILP. GAMS provides an interface with MATLAB.
6. Graph visualization tools Gephi [88] and GraphViz [89] are used to enable high-quality visualizations of medium and large-scale graphs.
7. Operation system command-line interface (CLI) is used to call the executable files of various graph partitioning tools (e.g., METIS [90], hMETIS [91], KaHIP [92]).

The described software interconnections allow to combine graph partitioning approaches of different nature to improve the quality and decrease the time of power sys-

tem area identification, to evaluate both static and dynamic performance of the obtained areas, and to visualize the results.

1.4. THESIS OUTLINE

The organization of the thesis is illustrated in Figure 1.2. The remainder of this section explains the contents of each chapter in more detail.

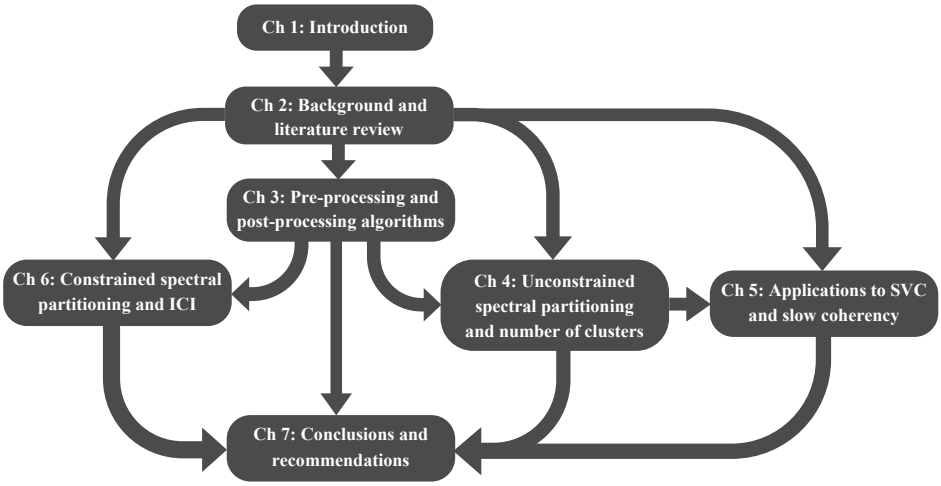


Figure 1.2: Thesis organization

Chapter 1 provides the background and motivation for this work. It explains the research challenges, the essential clustering terminology and the research approach.

Chapter 2 starts from an overview of the selected WAMPAC applications that are important for this thesis. It then continues with a literature review on graph partitioning, which is aimed to justify the focus on spectral approaches to partitioning. Next, the chapter provides an introduction into the theory of spectral graph partitioning and generator slow coherency, which are essential for the rest of the thesis. Chapter 2 also introduces the partitioning quality metrics relevant for this thesis.

Chapter 3 is dedicated to pre-processing and post-processing approaches that tackle the inherent flaws of graph partitioning. The proposed pre-processing solution is a graph outlier detection method that should help to avoid very small clusters. The first post-processing method aims to tackle the problem disconnected partitions that is very common to graph partitioning, while the second one aims to refine the partitioning with an efficient local search algorithm. Chapter 3 also describes some graph reduction techniques especially relevant for constrained graph partitioning.

Chapter 4 describes an efficient clustering algorithm suitable for various power system clustering tasks that do not include node grouping constraints. The proposed algorithm is based on ideas from spectral clustering, and its is able to estimate the optimal number of clusters, to ensure the area connectedness requirement, and to promote sufficiently large clusters.

Chapter 5 applies the ideas and algorithms detailed in Chapter 4 to two relevant applications that require power system subdivision into zones or areas. The initial framework in Chapter 4 is shown to produce good results for the task of adaptive zone division (AZD) for SVC. Subsequently, a modified grouping approach is proposed to improve the efficiency of power system slow coherency identification through spectral clustering.

Chapter 6 addresses constrained graph partitioning. The problem of finding trees spanning all the nodes of each constrained node group, which is also known as *packing Steiner trees* [77], is at the core of the chapter. A novel heuristic method is proposed to solve this problem in polynomial-time. After a comparison with some existing state-of-the-art alternatives, this constrained graph partitioning method is applied to find good initial solutions for a more accurate MILP-based ICI formulation.

Chapter 7 concludes this PhD research, and provides an outlook on possible research extensions and future work.

2

REVIEW OF AREA DEFINITION AND GRAPH PARTITIONING

2.1. INTRODUCTION

The goal of this chapter is to introduce the relevant theoretical background and power system applications. It starts with a more detailed description of SVC and ICI, highlighting the role of control areas and islands. Next, a classification of graph partitioning algorithms provides brief insights into the properties and advantages of the most common types of graph partitioning approaches, including the power system use cases. After introducing the adopted mathematical notation, the chapter explains the essentials of spectral clustering, which plays the key role in this thesis. A separate section is dedicated to the basics of model-based generator slow coherency approaches. Based on the criteria to power system areas listed in Section 1.2.2, a number of metrics is introduced to assess the quality of power network partitioning.

2.2. OVERVIEW OF SELECTED POWER SYSTEM APPLICATIONS

This section introduces the power system applications related to area-based control and protection that have motivated this research and are relevant for the case studies that illustrate the proposed methods.

2.2.1. SECONDARY VOLTAGE CONTROL

Secondary voltage control (SVC) is a part of the hierarchical voltage control that is implemented in several power grids around the globe (e.g., France, Italy, Romania, South African Republic, Republic of Korea, China) as a means to achieve a more robust voltage profile, reduce losses, increase power transfer capacity, increase reactive power reserves and voltage stability margins [51, 93]. Similarly to hierarchical AGC, a common hierarchical voltage control architecture typically consists of the primary, secondary and tertiary levels, as shown in Figure 2.1.

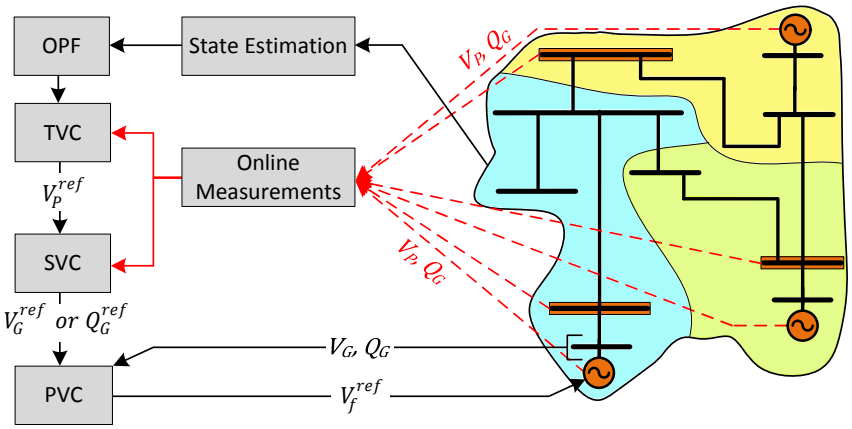


Figure 2.1: Hierarchical voltage control structure (adapted from [94]). Pilot buses are encircled in orange.

Primary voltage control (PVC) is the local voltage control that is implemented in any power system by the means of generator automatic voltage regulators (AVRs), on load tap changers (OLTCs), shunt reactors and capacitors, FACTS devices etc. The goal of PVC is to automatically respond to the voltage disturbances to maintain the voltage at the local busbar. The typical timeframe of PVC is in the order of a few seconds [15, 51]. PVC can be sufficient to ensure power quality and security when the power system load does not significantly change and there are no sudden large disturbances. However, as the system loading grows, many locally controlled reactive power resources may hit their reactive power limits and stop providing voltage support. Additionally, voltage regulation at power plants alone may not be enough to preserve the high voltage profile for the whole power system when the load shows a significant rapid increase. These simple observations illustrate some of the pitfalls of having only local voltage regulation without higher level coordination between the reactive power providers.

SVC resolves many issues of PVC by coordinating the control of several reactive power resources over electrically coherent voltage control zones (VCZs) [46, 47]. Typically, a large power system area such as a national or provincial power grid is subdivided into a number of VCZs featuring cohesive voltage profiles. Inside of each VCZ, a *pilot bus* is selected to control the zone's voltage. The pilot bus is typically an EHV bus whose voltage well represents the voltage of other zone's buses and is well controllable by the generators inside the zone. SVC maintains the voltage profile of its VCZ by adjusting the voltage setpoints of SVC control generators to counteract voltage deviations inside the zone by regulating the zone's pilot bus voltage to its reference value. If the neighbouring VCZ are not well decoupled, the inter-zone interactions can be handled (at least to some extent) by the control law. By its nature, SVC is a fully automatic feedback control system with the typical dominant time constant of several tens of seconds [15, 47]. The selection of pilot buses and associated control generators plays the crucial role in the SVC performance (and by this, in the overall performance of hierarchical voltage control). Clearly, not every combination of pilot buses is equally efficient. A good pilot bus selection ensures a significantly more robust system voltage profile compared to the case when only

PVC is active. Typically, both the optimal number and locations of pilot buses need to be determined, which represents a difficult combinatorial problem. So far, the two classes of methods are commonly used to tackle the pilot bus selection problem:

- The identification of VCZ followed by the selection of pilot buses inside of each VCZ using heuristics or direct enumeration [46, 47, 52].
- The use of greedy heuristic algorithms to select the initial set of pilot buses followed by local or global search algorithms to improve upon the initial selection [95, 96].

The first class of methods is adopted for this thesis. It has the advantage of not only guiding the search of pilot buses, but also highlighting their "areas of influence" (i.e., the VCZs). If the underlying VCZ identification algorithm is precise, a greater control over pilot bus selection can be achieved. At the same time, the use of greedy heuristic algorithms may suffer from local optima, and the subsequent direct use of search algorithms may result in only limited improvement.

SVC often receives the reference values of pilot bus voltages from a higher-level optimization program, which is based on a grid-wide OPF with the objective to minimize losses, increase reactive power margins, or maintain power system security under a set of contingencies. Such OPF computation acts as a short-term forecasting and is not a part of closed-loop control [51]. Some power systems (e.g., Italy) additionally implement TVC as the third level of hierarchical closed-loop voltage control. The goal of TVC is to update the SVC setpoints through a real-time grid-wide optimization based on the real-time values of pilot bus voltages and their forecasted values by the OPF [51]. The dominant time constant of TVC is in the order of several minutes [51].

2.2.2. INTENTIONAL CONTROLLED ISLANDING

Intentional controlled islanding is a novel emergency control technique, the purpose of which is to mitigate wide-area instabilities by intelligently separating the power network into a set of self-sustainable islands. This emergency control action can be used to isolate different kinds of adverse scenarios in power systems, e.g. loss of synchronism, cascading trips, voltage collapse or undamped oscillations. As compared to the traditional SIPS, such as out-of-step protection, ICI is an adaptive real-time emergency control algorithm that aims to consider multiple objectives when separating the network. During the last decades, it has gained an increased attention due to the recent severe blackouts all over the world (cf. Section 1.1.1). Given its nature (i.e., the last resort for blackout prevention), ICI must be adopted as quickly as possible. A typical sequence of events leading to the execution of ICI and the ICI-related actions are shown in Figure 2.2.

In Figure 2.2, the upper timeline illustrates the typical development of a power system blackout. The common reliability guidelines state that the power system should normally be operated with *N-1 redundancy* (i.e., a failure of any single component should not lead to the violation of the system operating limits for the normal condition). Therefore, a typical blackout starts from an initiating event that is followed by a sequence of outages (steady-state progression), during which some remedial actions (RAs) may be taken. Typical RAs are transmission switching, reactor switching, generator re-dispatch,

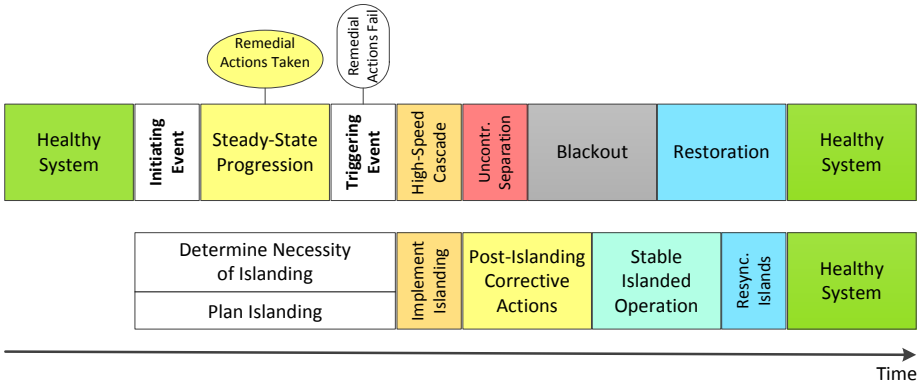


Figure 2.2: Typical sequence of events leading to ICI (adapted from [59]). The upper timeline shows the events in the absence of ICI. The lower timeline shows the events if ICI is successfully implemented.

or load shedding, and these actions can be event-driven (e.g., fast load shedding in response to a failure of a critical line). If RAs do not stop the outages, one of the outages will constitute the triggering event for the loss of stability (e.g., uncontrolled separation or voltage collapse), which is followed by a complete or partial blackout.

To avoid the blackout scenario, the ICI program should continuously monitor the power system to determine the necessity of islanding and to plan islanding. Once it becomes clear that the system has become unstable, the most recent islanding solution should be implemented to isolate the faulted area and prevent the contagion of the rest of the network. This reasoning gives rise to the following two key questions of ICI [19]:

- *When to island?* That is, how to reliably detect instability and to choose the proper moment to separate the network to ensure transient stability during islanding.
- *Where to island?* That is, which transmission lines to open to form islands that are steady-state stable, satisfy the system operating limits for the emergency condition and require a minimum amount of load shedding.

From the above two questions, the first one is the problem of ICI timing, which is more related to power system dynamic stability and online DSA. The second question is the problem of ICI switching that is relevant for this thesis, as it represents a network partitioning problem with multiple power system related constraints. The important constraints associated with the second ICI question are listed below:

1. Island connectedness. Each island should represent a connected network area.
2. Generator coherency. The separated islands close to the unstable area should include generators with coherent rotor angle dynamics as the means to promote their transient stability and synchronization [35, 43, 55, 58].
3. Blackstart unit availability. Each formed island should include at least one blackstart power source to restore the island if it collapses.

4. Transmission line availability. Not every power network branch can be opened to form islands, but only transmission lines with synchro-check relays that would enable the resynchronization of islands [59]. In particular, transformer branches cannot be opened for emergency control actions.
5. Load-generation balance. The maximal generation capacity of an island should closely match the island's load.
6. Equipment operational limits. Following the ICI, power system elements inside of the islands should be able to operate for an extended period of time.
7. Static voltage stability. Each formed island should have a feasible AC load flow solution with a sufficient voltage stability margin.
8. Frequency stability. Each formed island should possess a sufficient inertia to prevent the rapid frequency decline following the separation.

From the above requirements, the first four can be stated in terms of MILP constraints, which together represent a constrained graph partitioning problem. The generator coherency requirement can be cast as a node grouping constraint once the coherent generator groups to be included into each island are known. The blackstart unit availability requirement is similar, as it demands certain blackstart units to be assigned to a certain island. The transmission line availability requirement actually simplifies the problem, as it allows to reduce the power network branches that cannot be open. The last four requirements are based on the power flow and swing equations and represent the power system specific limitations on the graph partitioning problem. Thus, the question *"Where to island?"* can be formulated as a constrained graph partitioning problem subject to physical power system constraints.

2.3. OVERVIEW OF GRAPH PARTITIONING METHODS

The introductory overviews in Chapter 1 and Section 2.2 should explain the meaning of area identification to power systems and the role of graph partitioning in power system area identification. The goal of this section is to provide a helicopter view on graph clustering and partitioning.

Data clustering using graph models is a popular approach in many technical domains. A great variety of specialized graph clustering algorithms is used in such disciplines as data mining, pattern recognition, complex network analysis, and bioinformatics (e.g., some of the notable references are [68, 97, 98, 99, 100, 101, 102]). Due to this large variety, it seems prudent to focus on the groups of methods, and largely dismiss the algorithms that are hard to assign to a particular group. In Figure 2.3, a possible classification of graph clustering algorithms is proposed, which is mainly based on [66], [103] and [104]. The classification in Figure 2.3 aims to categorize the main working principles of various clustering algorithms into eight classes, and for certain classes the representative class members are given in the third level of the classification hierarchy (shown in a darker blue color). A concrete graph clustering algorithm can combine several working principles (e.g., many multilevel partitioners make use of spectral methods and local search [90, 92]), so the representative algorithms in Figure 2.3 are assigned based on what is perceived as their main working principle.

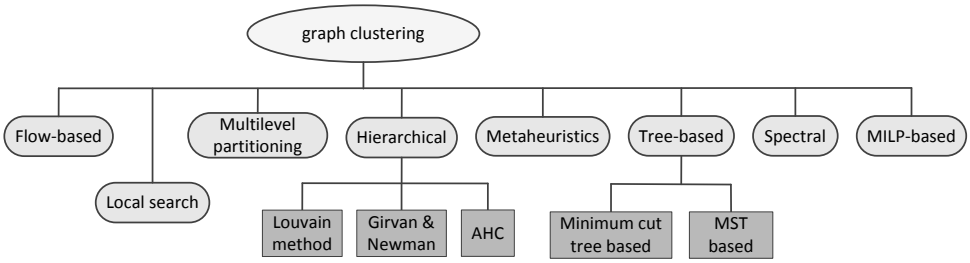


Figure 2.3: Classification of graph clustering algorithms

Multilevel graph partitioning is a successful graph partitioning framework that generally consists of coarsening and refinement phases. The coarsening phase corresponds to a series of reductions of the initial graph aiming to achieve a small graph that still largely characterizes the initial cluster structure. At the coarsest level, the graph is partitioned with a simple algorithm like spectral recursive bisection [105]. During the refinement phase, the coarsening process is reversed. The partitioning is mapped from coarser levels to finer ones and then refined by using graph cut improvement algorithms (e.g., Fiduccia-Mattheyses [106]). In the context of electric power networks, the multilevel partitioning software METIS has been applied to the ICI problem in [58, 107].

Hierarchical graph clustering aggregates multiple algorithms based on bottom-up merging (e.g., the Louvain method [100]) or top-down splitting (e.g., the Girvan–Newman algorithm [101]) of network clusters. The bottom-up (or agglomerative) variety usually starts with every node belonging to a separate cluster and proceeds by combining together the most similar clusters until all nodes are combined into one cluster. The top-down variety usually starts with the input graph assigned to a single cluster, which is then sequentially subdivided until each node becomes a separate cluster. The both varieties may terminate early if some predefined criteria are met. Hierarchical clustering approaches are quite common to electric power systems, with the use cases including control zone definition [46] and structural analysis of power grids [76].

Tree-based graph clustering can be seen as a variety of hierarchical graph clustering, with the notable difference that it operates on a tree representing the input graph. This tree can be a minimum spanning tree (MST) as in MST-based clustering [99] or a minimum cut tree [102]. There exist both top-down and bottom-up varieties of tree-based graph clustering.

Flow-based graph clustering uses the theory of network flows [108] to cluster or partition the graph. Given a set of *terminal nodes* in the graph, an approximate method to separate each terminal node from all others that utilizes network flows through the max-flow/min-cut theorem was proposed in [109]. The theory of network flows and max-flow/min-cut theorem are also highly valuable if terminal nodes are not present, and especially in situations when graph bi-partitioning is required [110].

Local search is a group of algorithms that accept an initial crude graph partition and try to improve it with respect to some objective function and constraints by performing incremental node swaps between the graph parts. Local search algorithms were among the earliest proposed approaches to partition a graph (e.g., the Kernighan–Lin heuristic

[111]), and they play an important role in solving practical graph partitioning problems that arise in microchip design and scientific computations.

MILP is the discrete optimization framework that offers a natural way to exactly state the majority of relevant graph partitioning problems (including the power system related ones). With a few exceptions (e.g., a totally unimodular constraint matrix), exact MILP problem formulations are NP-hard to solve. This implies long solution times on larger problem instances and general lack of guarantee to find a solution within a prescribed time limit. Coincidentally, the exact optimal area structure is often not essential, and not all issues related to area identification can be easily formulated as MILP (e.g., the optimum number of areas). The combination of these factors limits the relevancy of MILP in many practical area identification studies, although MILP was applied to ICI [112, 113], PPSR [114] as well as to many planning and scheduling problems for which the solution time is not critical [11, 62, 64].

Metaheuristic optimization algorithms represent a diverse group of methods that are exploring the search space of an optimization problem using efficient heuristic rules to obtain good or near-optimal solutions. The optimization problem at hand can be very general (i.e., highly nonlinear, non-convex, mixed-integer etc). Metaheuristic optimization is typically much slower than proper clustering algorithms (i.e., hierarchical, spectral etc), but it may still be faster than MILP at finding good feasible solutions. Metaheuristic algorithms are often applied to power systems, including flexible definition of control areas [30], partitioning for parallel processing [115], and ICI [116].

Spectral graph clustering is a family of methods based on spectral graph theory [73]. Spectral clustering algorithms are among the most well-established methods for graph clustering and partitioning, which is due to their long history [105], strong mathematical foundation [73, 74], and numerous extensions, including different objective functions and clustering with node label information [57, 69]. Spectral clustering algorithms rely on calculating first several eigenvectors of certain graph matrices, which is an efficient computation implemented in several high-performance computing libraries (e.g., ARPAC). In the context of power systems, spectral clustering has been applied to clustering of contingencies [117], clustering of load profiles [118, 119], structural analysis of power networks [76], AZD for SVC [**J1**, 94], PPSR [45], ICI [43, 61], grid partitioning for parallel processing [120], power network reduction [37, 67].

To conclude this section, it is worth to mention that a number of specialized power system clustering methods can be considered as graph clustering. For example, Chow [35, 121] proposed a tolerance-based coherency grouping algorithm based on generator distance matrix, and Corsi [47, 51] proposed a tolerance-based algorithm for VCZ identification based on voltage sensitivity matrix. The outcomes of the both algorithms vary depending on the selected values of tolerance parameters (larger tolerances correspond to looser clusterings), which may become evident after looking at the algorithm descriptions in [35, 47, 51, 121]. More importantly, both the distance matrix in [35, 121] and the sensitivity matrix in [51] can been seen as the adjacency matrices of weighted undirected full graphs of generator distances or voltage sensitivities. The discussed methods are of practical importance, as the ERPI's DYNRED program includes the grouping algorithm [121], and the VCZ division of the Italian SVC implementation is computed with the zoning algorithm [51].

2.4. MATHEMATICAL NOTATION

Prior to exploring the relevant theoretical background, it is useful to introduce the mathematical notation used throughout the thesis.

Matrix and Vector Notation. In this thesis, matrices are typeset in bold uppercase font (e.g., $\mathbf{A}$), vectors are typeset in bold italic font (e.g., $\boldsymbol{V}$ or $\boldsymbol{\delta}$), and scalar variables are typeset in italic font (e.g., X or x). Sizes of matrices are denoted by subscripts (e.g., $\mathbf{A}_{n\times k}$ for a matrix with n rows and k columns) or introduced when the matrix is defined (e.g., $\mathbf{A} \in \mathbb{R}^{n\times k}$). A matrix element is discriminated by its row and column indices written in the subscript (e.g., a_{ij}). To denote an all-ones matrix, $\mathbf{1}$ is used, and an all-zeros matrix is denoted by $\mathbf{0}$. The identity matrix of size k is denoted as $\mathbf{I_k}$. A diagonal matrix formed from a vector argument is denoted as diag$(\cdot)$. A block diagonal matrix formed from a list of matrices is denoted as blkdiag$(\cdot)$. A column vector consisting of subsequent elements on the main diagonal of a square matrix is denoted as Diag$(\cdot)$. To avoid an excessive stacking of matrix indices, an alternative matrix indexing is occasionally used in the algorithm pseudocodes (e.g., $\mathbf{A}[i,j]$ equals to a_{ij}). Submatrices are occasionally used in pseudocodes, where they are denoted by using the same bracket notation (e.g., $\mathbf{A}[1,\ldots,n;1,\ldots,3]$ is the submatrix formed by the first n rows and three columns of $\mathbf{A}$).

Graph Notation. An undirected graph is denoted as $\mathcal{G} = (\mathcal{V},\mathcal{E})$. Its node set $\mathcal{V}$ is denoted as $\mathcal{V} = \{v_1,\ldots,v_n\}$ and the edge set is denoted as $\mathcal{E} \subset \mathcal{V}\times\mathcal{V}$. The graph $\mathcal{G}$ may have node weights $\mu(v) \in \mathbb{R}_+$ for each $v \in \mathcal{V}$. The node weights generally express the "importance" of a node, with larger weights meaning higher importance. The graph $\mathcal{G}$ may also have edge weights $W(e) \in \mathbb{R}_+$ for each $e \in \mathcal{E}$, and they can express either distance or similarity between the edges' endnodes. The symmetric adjacency matrix of $\mathcal{G}$ containing the edge weights is given by $\mathbf{W} = [W_{ij}]$ ($\mathbf{W} \in \mathbb{R}^{n\times n}$), and the rectangular unweighted incidence matrix associated with $\mathcal{G}$ is given by $\mathbf{C} = [c_{ij}]$ ($\mathbf{C} \in \mathbb{R}^{n\times m}$). The numeric indices of the rows and columns of the adjacency matrix $\mathbf{W}$ are derived from the node subscripts in the set $\mathcal{V}$. A set of nodes that should be kept together in a separate connected cluster (i.e., a node grouping constraint) is denoted as $\mathcal{T}$, as such nodes are often called *terminal nodes* in the literature [77, 108, 109]. Constrained graph partitioning problems are specified by defining a set $\{\mathcal{T}_1,\ldots,\mathcal{T}_k\}$ of node grouping constraints, and possibly some other constraints.

> From this point and further on, the terms *bus* and *node* as well as *branch* and *edge* can be used interchangeably to refer to the analogous concepts in the power network and its graph representation.

Partitioning Notation. Following Luxburg [74], $\mathbb{1}_C$ defines the indicator vector of the nodes belonging to the cluster or graph connected component $\mathcal{C}$. That is, $\mathbb{1}_C = [f_1,\ldots,f_n]^T$ and $f_i = 1$ if node i belongs to $\mathcal{C}$ and $f_i = 0$ otherwise. For a graph partitioned into k blocks, it is possible to define a *partition indicator matrix* as a concatenation of blocks' indicator vectors: $[\mathbb{1}_{C_1},\ldots,\mathbb{1}_{C_k}]$. Partition indicator matrices are marked with a tilde (e.g., $\widetilde{\mathbf{A}}$).

Other Notation. Sets are typeset in uppercase calligraphic font (e.g., $\mathcal{E}$). Final solutions of optimization algorithms are marked with an asterisk (e.g., $\mathbf{R}^*$). The standard big-O notation is used to describe the time complexity of algorithms.

2.5. SPECTRAL CLUSTERING

As it has already been revealed in Section 1.3, spectral clustering based approaches play the key role for the developments of this thesis, which warrants including a separate section about their essentials. Spectral clustering is a family of methods based on computing special eigenvalue problems involving graph matrices. This section is first going to describe its basic setting based on [74, 122, 123]. The two spectral clustering algorithms that are going to be used for benchmarking and computational experiments are detailed in the second part of this section.

2.5.1. THEORETICAL BACKGROUND

Spectral clustering is a similarity-based clustering framework; that is, the graph edge weights should correspond to a quantity representing the closeness or similarity between a pair of nodes (e.g., a power flow or a series admittance). Assuming that the adjacency matrix $\mathbf{W}$ of the input graph $\mathcal{G}$ is a *similarity matrix*, let us define the *(weighted) degree* of node $v_i \in \mathcal{V}$:

$$D_i = \sum_{j=1}^{n} W_{ij} \tag{2.1}$$

For similarity graphs, the weighted node degree (2.1) can be interpreted as the weight of node v_i, as well-connected nodes of similarity graphs have high weighted degrees. The *degree matrix* is defined from (2.1) as:

$$\mathbf{D} = \mathrm{diag}(D_1, \ldots, D_n) \tag{2.2}$$

Next, the sum of weights of the edges running between two sets of nodes $\mathcal{A}$ and $\mathcal{B}$ is introduced as [124]:

$$\mathrm{links}(\mathcal{A}, \mathcal{B}) = \sum_{i \in \mathcal{A}, j \in \mathcal{B}} W_{ij} \tag{2.3}$$

In particular, the sum of node degrees of cluster $\mathcal{C}$ can be expressed in terms of (2.3) as $\mathrm{links}(\mathcal{C}, \mathcal{V})$, which is denoted as cluster volume [74]:

$$\mathrm{vol}(\mathcal{C}) = \sum_{i \in \mathcal{C}} D_i \tag{2.4}$$

The sum of edge weights of the edges bordering cluster $\mathcal{C}$ can be expressed in terms of (2.3) as $\mathrm{links}(\mathcal{C}, \mathcal{V} \setminus \mathcal{C})$, which is denoted as cluster cut [74]:

$$\mathrm{cut}(\mathcal{C}) = \sum_{i \in \mathcal{C}, j \in \mathcal{V} \setminus \mathcal{C}} W_{ij} \tag{2.5}$$

Based on the above definitions, the following matrices are defined [74]:

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \tag{2.6a}$$

$$\mathbf{L_{rw}} = \mathbf{D}^{-1}\mathbf{L} \tag{2.6b}$$

$$\mathbf{L_n} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}} \tag{2.6c}$$

where $\mathbf{D}^{-\frac{1}{2}} = \text{diag}\left(\frac{1}{\sqrt{D_1}}, \ldots, \frac{1}{\sqrt{D_n}}\right)$. Matrix (2.6a) is called *graph Laplacian*. It has a great theoretic significance, but it is rarely used in this thesis. The other two matrices (2.6b)–(2.6c) are called *random walk normalized Laplacian* and *symmetric normalized Laplacian* respectively. Their eigenvalues and eigenvectors share the following important properties [74, 76]:

1. The eigenvalues of $\mathbf{L_{rw}}$ are real and satisfy the inequality $0 \leq \nu_i \leq 2$, $i = 1, \ldots, n$;
2. 0 is an eigenvalue of $\mathbf{L_{rw}}$, and its multiplicity is equal to the number of connected components of $\mathscr{G}$; the eigenspace of 0 is spanned by the k indicator vectors of connected components $\mathbb{1}_{C_j}$, where $\mathcal{C}_1, \ldots, \mathcal{C}_k$ represent the k connected components of $\mathscr{G}$;
3. $\mathbf{L_{rw}}$ and $\mathbf{L_n}$ have the same eigenvalues. The corresponding eigenvectors relate as $\boldsymbol{U} = \mathbf{D}^{\frac{1}{2}}\boldsymbol{V}$, where $\boldsymbol{V}$ is the eigenvector of $\mathbf{L_{rw}}$ and $\boldsymbol{U}$ is the eigenvector of $\mathbf{L_n}$ both corresponding to the same eigenvalue.

The proofs of these properties can be found in [73, 74], and they can be in part deduced by noticing that $\mathbf{L_{rw}}$ and $\mathbf{L_n}$ are similar matrices.

The matrices (2.6b)–(2.6c) are important for clustering due to the existing possibility to express the important Ncut criterion through them. Although finding a minimum cut separating the graph in two parts can be achieved in the $O(|\mathcal{E}||\mathcal{V}| + |\mathcal{V}|^2 \log|\mathcal{V}|)$ time [125], the resulting partitions are often very unbalanced up to the point that the smaller partition may consist of a single node. Additionally, graph minimum cuts may be not robust with respect to edge weights perturbations; that is, a small change of the graph edge weights may lead to a significant change of the minimum graph cut. These issues prevent the direct use of the exact minimum graph cut algorithms for the clustering purposes. The balanced graph cuts [66, 126] circumvent the drawbacks of minimum graph cuts by balancing the cluster cut (2.5) by the cluster size, which in case of the Ncut criterion is given by the cluster volume (2.4):

$$\text{Ncut}(\mathcal{C}_1, \ldots, \mathcal{C}_k) = \frac{1}{k}\sum_{j=1}^{k}\frac{\text{cut}(\mathcal{C}_j)}{\text{vol}(\mathcal{C}_j)} \tag{2.7}$$

Thus, minimizing the Ncut criterion aims at finding graph clusters that are well-separated (i.e., low cluster cuts), while not too small (i.e., cluster volumes are large enough to not cancel out the low cluster cut values). Unfortunately, minimizing balanced cluster cuts is an NP-complete problem even for $k = 2$ on planar graphs [74, 127]. In such situations it is common to search for approximations of the original problem that can be solved fast by neglecting some difficult constraints of the original problem. To start looking for approximations, the Ncut minimization problem is first expressed in the matrix form [122, 123]:

$$\text{minimize} \quad \text{Ncut}(\widetilde{\mathbf{X}}) = \frac{1}{k}\sum_{j=1}^{k}\frac{\widetilde{\boldsymbol{X}}_j^T\mathbf{L}\widetilde{\boldsymbol{X}}_j}{\widetilde{\boldsymbol{X}}_j^T\mathbf{D}\widetilde{\boldsymbol{X}}_j} \tag{2.8a}$$

$$\text{subject to:} \quad \widetilde{\mathbf{X}} \in \{0,1\}^{n\times k}, \ \widetilde{\mathbf{X}}\mathbf{1}_{k\times 1} = \mathbf{1}_{n\times 1} \tag{2.8b}$$

where $\widetilde{\mathbf{X}} = [\widetilde{\boldsymbol{X}}_1, \ldots, \widetilde{\boldsymbol{X}}_k]$ is the partition indicator matrix of clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$ that should lead to the optimal Ncut value, and $\widetilde{\boldsymbol{X}}_j$ are the indicator vectors of the individual clusters that need to be found. The nominator of (2.8a) is $\text{cut}(\mathcal{C}_j)$ written in the matrix form, and the denominator of (2.8a) is $\text{vol}(\mathcal{C}_j)$ written in the matrix form, with $j = 1, \ldots, k$. The first constraint of (2.8b) is requiring each indicator vector to take discrete values, and the second constraint is forcing the indicator vectors to be orthogonal and mutually exclusive.

The authors of [122] perform the change of variables in (2.8) by introducing a *scaled partition matrix* $\widetilde{\mathbf{V}}$:

$$\widetilde{\mathbf{V}} = \widetilde{\mathbf{X}}(\widetilde{\mathbf{X}}^T \mathbf{D} \widetilde{\mathbf{X}})^{-\frac{1}{2}} \tag{2.9}$$

The matrix $\widetilde{\mathbf{V}}$ scales the binary indicator columns of $\widetilde{\mathbf{X}}$ by the inverse square root volumes of partitions [122], which is consistent with the form of Ncut indicator vectors given in [74]. Next, the original constraints of (2.8) are relaxed and the change of variables is performed, which reduces problem (2.8) to a trace minimization problem [122, 123]:

$$\text{minimize} \quad \text{Ncut}^{\text{SR}}(\mathbf{V}) = \frac{1}{k} \text{tr}\left(\mathbf{V}^T \mathbf{L} \mathbf{V}\right) \tag{2.10a}$$

$$\text{subject to:} \quad \mathbf{V}^T \mathbf{D} \mathbf{V} = \mathbf{I_k} \tag{2.10b}$$

or equivalently, through another change of variables $\mathbf{V} = \mathbf{D}^{-\frac{1}{2}} \mathbf{U}$:

$$\text{minimize} \quad \text{Ncut}^{\text{SR}}(\mathbf{U}) = \frac{1}{k} \text{tr}\left(\mathbf{U}^T \mathbf{L_n} \mathbf{U}\right) \tag{2.11a}$$

$$\text{subject to:} \quad \mathbf{U}^T \mathbf{U} = \mathbf{I_k} \tag{2.11b}$$

where $\text{tr}(\cdot)$ is the matrix trace, the SR addition to the objective name stands for *Spectral Relaxation*, tildes above the variables are removed due to the neglected discreteness constraints (2.8b), and constraints (2.10b), (2.11b) come naturally through algebraic reduction.

The relaxed Ncut minimization problem (2.11) can be directly solved using the Rayleigh-Ritz theorem and its extensions (see the Appendix of [123] for the proofs). Namely, an optimal solution to (2.11) is given by the matrix formed by the first k eigenvectors of $\mathbf{L_n}$, and the optimal solution value is given by the sum of the first k eigenvalues of $\mathbf{L_n}$ [122]:

$$\text{Ncut}^{\text{SR}*} = \frac{1}{k} \sum_{j=1}^{k} \nu_j \tag{2.12a}$$

$$\mathbf{U}^* = [\boldsymbol{U_1}, \ldots, \boldsymbol{U_k}] \tag{2.12b}$$

where $\nu_1 \leq \ldots \leq \nu_n$ are the eigenvalues of $\mathbf{L_n}$ ordered nondecreasingly, and $\boldsymbol{U_1}, \ldots, \boldsymbol{U_n}$ are the corresponding eigenvectors.

According to the Rayleigh-Ritz theorem, the Ncut^{SR} problem stated as (2.10) has the eigenvectors of $\mathbf{L_{rw}}$ as its optimal solution $\mathbf{V}^*$ (the optimal objective values of (2.10) and (2.11) are the same, as the eigenvalues of $\mathbf{L_{rw}}$ and $\mathbf{L_n}$ are equal). However, it is more computationally efficient to compute eigenvectors of symmetric matrices. Thus, the eigenvectors of the symmetric normalized Laplacian $\mathbf{L_n}$, which are related to the eigenvectors

of $\mathbf{L_{rw}}$ by the abovementioned properties of graph Laplacian matrices, are more beneficial to compute.

The presented derivation of the spectral relaxation of the Ncut criterion (2.7) uses the matrices $\mathbf{L_{rw}}$ and $\mathbf{L_n}$, which are common across the literature. However, it is also possible to use the *random walk transition matrix* (2.13a) and the *normalized adjacency matrix* (2.13b) for the same purpose (see [122]).

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{W} \tag{2.13a}$$

$$\mathbf{W_n} = \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}} \tag{2.13b}$$

The random walk transition matrix $\mathbf{P}$ relates to the random walk normalized Laplacian as $\mathbf{L_{rw}} = \mathbf{I} - \mathbf{P}$, and the normalized adjacency matrix $\mathbf{W_n}$ is linked with the normalized symmetric Laplacian $\mathbf{L_n}$ as $\mathbf{L_n} = \mathbf{I} - \mathbf{W_n}$. These relationships imply that the *smallest* eigenvalues of $\mathbf{L_n}$ and $\mathbf{L_{rw}}$ correspond to the *largest* eigenvalues of $\mathbf{W_n}$ and $\mathbf{P}$. That is, the eigenvalues $0 \leq \nu_i \leq 2$ of $\mathbf{L_n}$ and $\mathbf{L_{rw}}$ correspond to the eigenvalues $1 \leq \lambda_i \leq -1$ of $\mathbf{W_n}$ and $\mathbf{P}$ for $i = 1, \ldots, n$. For the shown correspondence in the eigenvalues, $\mathbf{P}$ has the same eigenvectors as $\mathbf{L_{rw}}$, and $\mathbf{W_n}$ has the same eigenvectors as $\mathbf{L_n}$.

As the eigenvectors of $\mathbf{L_n}$ and $\mathbf{W_n}$ are equivalent, they can be obtained more efficiently by computing the eigendecomposition of the real symmetric normalized adjacency matrix $\mathbf{W_n}$ (2.13b). The computation of eigenvectors of $\mathbf{W_n}$ is more efficient due to the properties of the state-of-the-art numerical algorithms: computing first k *largest* eigenpairs of a sparse matrix with iterative eigensolvers (e.g., the Lanczos method [128]) has better numerical properties than computing first k smallest eigenpairs. Thus, the matrix $\mathbf{W_n}$ is mostly going to be used for the computation of eigenvectors for spectral clustering (except the topics related to Section 5.3). The normalized adjacency matrix has the following properties:

1. The eigenvalues of $\mathbf{W_n}$ are real and satisfy the inequality $-1 \leq \lambda_i \leq 1$, $i = 1, \ldots, n$;
2. 1 is an eigenvalue of $\mathbf{W_n}$, and its multiplicity is equal to the number of connected components of $\mathscr{G}$; the eigenspace of 1 is spanned by the k column vectors $\mathbf{D}^{\frac{1}{2}}\mathbf{1}_{C_j}$, where $C_1, \ldots, C_k$ represent the k connected components of $\mathscr{G}$.

As it can been seen, the eigenvalue properties of the matrix $\mathbf{W_n}$ are very similar to those of the matrices $\mathbf{L_{rw}}$ and $\mathbf{L_n}$, which is due to their close relatedness.

The first k smallest eigenvectors of $\mathbf{L_n}$ or the first k largest eigenvectors of $\mathbf{W_n}$ can be combined into the matrix $\mathbf{U} \in \mathbb{R}^{n \times k}$, see (2.12b). The rows of $\mathbf{U}$ can be seen as the coordinates of the nodes of the original power network in $\mathbb{R}^k$. This representation of the nodes of the original network by the points in the Euclidean space formed by the first k eigenvector coordinates is often called *spectral embedding* of the network in $\mathbb{R}^k$ [74, 76].

The second property of $\mathbf{W_n}$ provides an additional view on the use of its largest eigenvectors for clustering purposes (i.e., in addition to the Ncut approximation view). If $\mathscr{G}$ has k connected components, the rows of $\mathbf{U}$ will lie along the axes of the canonical coordinate system in $\mathbb{R}^k$ (see the second property of $\mathbf{W_n}$), and the k connected components will be easily retrievable from $\mathbf{U}$. The multiple connected components of $\mathscr{G}$ can also be considered as perfectly separated clusters. According to matrix perturbation theory [74], the addition of some low-weight edges between the k perfectly separated clusters only

slightly perturbs the k largest eigenvectors from their ideal values. Thus, an observation can be made [74, 79, 129] that the more the first k eigenvectors resemble the ideal structure corresponding to fully separated clusters, the more closely those eigenvectors represent the dominant clustering structure of $\mathscr{G}$.

2.5.2. SPECTRAL CLUSTERING ALGORITHMS

The most common final step of spectral clustering is to assign each row of an eigenvector matrix (e.g., $\mathbf{U}$) to a fixed cluster. This procedure is commonly referred to as *discretization*, and its result can be thought of as a conversion of real-valued eigenvector matrix into a *discrete* matrix $\widetilde{\mathbf{X}} \in \{0,1\}_{n\times k}$ with the property $\widetilde{\mathbf{X}}\mathbf{1}_{k\times 1} = \mathbf{1}_{n\times 1}$ (cf. (2.8)).

In practice, the rows of the eigenvector matrix $\mathbf{U}$ are often normalized to have length one [74, 79]. Therefore, it is convenient to introduce the matrix $\mathbf{X} \in \mathbb{R}^{n\times k}$ that is obtained from $\mathbf{U}$ by normalizing the rows of $\mathbf{U}$ to have length one.

$$X_{ij} = U_{ij}/(\textstyle\sum_{j=1}^{k} U_{ij}^2)^{1/2} \tag{2.14}$$

The procedure (2.14) is equivalent to projecting the point coordinates in $\mathbb{R}^k$ defined by the rows of matrix $\mathbf{U}$ onto the unit hypersphere centered at the origin of the same k-dimensional Euclidean space. These projected node coordinates are often referred to as *normalized spectral embedding*. The normalized spectral embedding was introduced and popularized by Ng et al. in [79]. The relevant features of the method [79] are summarized in Algorithm 2.1.

Algorithm 2.1 Spectral clustering according to Ng, Jordan, and Weiss

Input: Matrix $\mathbf{W_n}$, number of clusters k

1: $[\boldsymbol{U}_1, \ldots, \boldsymbol{U}_k] \leftarrow$ the first k eigenvectors of $\mathbf{W_n}$ in decreasing order of their eigenvalues.
2: $\mathbf{U} \leftarrow [\boldsymbol{U_1}, \ldots, \boldsymbol{U_k}]$
3: Form the matrix $\mathbf{X}_{n\times k}$ from $\mathbf{U}$ by normalizing the rows of $\mathbf{U}$ as (2.14).
4: Consider the n rows of $\mathbf{X}$ as the n points in $\mathbb{R}^k$ and cluster them into k clusters using k-means or any other algorithm that attempts to minimize cluster distortion.
5: Finally, assign the original node v_i to cluster $\mathcal{C}_j$ if and only if row i of the matrix $\mathbf{X}$ was assigned to cluster j by the clustering algorithm at the previous step.

Output: Clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$

Spectral clustering with Algorithm 2.1 is often referred to as the spectral clustering algorithm of Ng, Jordan, and Weiss (NJW) by the names of its authors, and it can also be seen as an eigenvector discretization procedure. Despite its popularity, the NJW clustering algorithm has an important drawback of not accounting for the graph interconnection structure that is encoded in the matrix $\mathbf{W}$ or $\mathbf{W_n}$. As the result, it is possible to obtain disconnected network clusters consisting of several graph connected components.

Assuming that $\mathbf{W_n}$ is readily available, the time complexity of Algorithm 2.1 is dominated by the iterative eigensolver routines for sparse matrices that have a complex convergence behavior depending on multiple factors [70], but tend to show a subquadratic run time in the size of of the input matrix n.

One way to resolve the cluster connectedness issue is to define the distances between the nodes not as Euclidean distances in spectral embedding, but as *shortest path distances* [130] in the graph $\mathcal{G}$ equipped with new edge weights that are equal to the Euclidean distance in spectral embedding between the end nodes of each edge. The input graph $\mathcal{G}$ equipped with the new set of edge weights derived from distances in spectral embedding is further referred to *spectrally embedded graph*. If the spectral embedding is normalized as (2.14) to lie on the unit hypersphere, spherical distances are more appropriate and can be used instead of the Euclidean ones [76]. However, it was noticed that using the spherical distances instead of Euclidean ones usually has a negligible influence on the partitioning results.

By using the shortest path distances in the spectrally embedded graph, both the closeness of graph nodes in spectral embedding and the graph interconnection structure are taken into account. However, with distances between graph nodes as an input, the popular centroid-based clustering algorithms like k-means cannot be used to cluster the spectral embedding. The solution proposed in [76] is to use the agglomerative hierarchical clustering (AHC) that can take a distance matrix between the data objects as its input. Providing only the distance matrix is possible with some of the linkage criteria of AHC, such as single linkage (SLINK), complete linkage (CLINK) or average linkage (UPGMA), which are described in many books on data science (e.g., [71]). The ability of the method [76] to ensure cluster connectedness warrants its use as a benchmark algorithm, the main ideas of which are summarized in Algorithm 2.2, which is further referred to as hierarchical spectral clustering (HSC).

Algorithm 2.2 Hierarchical spectral clustering

Input: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, matrix $\mathbf{W_n}$, number of clusters k

1: $[\boldsymbol{U}_1, \ldots, \boldsymbol{U}_k] \leftarrow$ the first k eigenvectors of $\mathbf{W_n}$ in decreasing order of their eigenvalues.
2: $\mathbf{U} \leftarrow [\boldsymbol{U_1}, \ldots, \boldsymbol{U_k}]$
3: Form the matrix $\mathbf{X}_{n\times k}$ from $\mathbf{U}$ by normalizing the rows of $\mathbf{U}$ as (2.14).
4: **for all** $e = (i, j) \in \mathcal{E}$ **do** $W_{sp}(e) = \sqrt{\sum_{l=1}^{k}(X_{il} - X_{jl})^2}$ **end for**
5: **DM** $\leftarrow$ Shortest paths distance matrix of $\mathcal{G}$ with respect to W_{sp}.
6: Perform AHC on **DM**. Assign the graph node v_i to cluster $\mathcal{C}_j$ if and only if v_i was assigned to cluster j by AHC on the graph distance matrix DM.

Output: Clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$

In Algorithm 2.2, the number of clusters k is assumed as an input, and the number of eigenvectors is assumed to be equal to the number of clusters. Although computing eigengaps as in [76] has connections to determining the optimal number of clusters (see [74]), it has been empirically observed that once k is decided, best clusterings are obtained from applying Algorithm 2.2 on the spectral embedding computed from first k eigenvectors of $\mathbf{W_n}$. The reason for this observation can be related to the interpretation of spectral embedding discussed in more detail in Chapter 4.

The run time of Algorithm 2.2 is dominated by the computation of the distance matrix **DM** in $O(|\mathcal{E}||\mathcal{V}| + |\mathcal{V}|^2 \log|\mathcal{V}|)$ and AHC, whose time complexity can be $O(n^2 log\, n)$ to $O(n^3)$ depending on the implementation, where n is the number of clustered objects.

2.6. GENERATOR SLOW COHERENCY

Generator coherency is defined as the property of a group of synchronous generators to maintain an approximately constant rotor angle difference after a disturbance [35, 131]. In other words, coherent generators react similarly when a disturbance occurs, or *swing together* following a disturbance. The coherent swings of large groups of generators at low oscillation frequencies are the manifestation of *slow coherency*; that is, of generator coherency with respect to the slow inter-area modes of power system [1, 34]. Thus, the identification of slow coherent generator groups helps to reveal the structure of large-scale power grids in terms of rotor angle dynamics.

> In what follows, the term *generator coherency* is going to be used with the meaning of *generator slow coherency*, as this is the only type of dynamic coherency relevant for this thesis. The terms *coherent generator group* and *coherent group* are both assumed to refer to a *slow coherent generator group*.

The major reason for this review of generator coherency is due to its role in ICI [43, 55, 113]. Namely, assigning non-coherent generators to the same island is likely to result in rotor angle instability for this island after the network separation. Generators forming a slow coherent group have a relatively strong dynamic coupling between each other, which ameliorates generator synchrony [35, 83]. Therefore, it is reasonable to utilize slow coherency identification approaches to find generators which should be grouped together for the purpose of ICI. In graph partitioning terms, generator coherency provides the essential node grouping constraints to ICI.

The known generator coherency identification methods can be broadly classified into model-based [33, 34, 121] and signal-based approaches [50, 132]. The signal-based approaches estimate generator coherency based on the measured signals of generator terminal voltage angles, frequencies or rotor angles. They typically require little to no knowledge about the power system structure and parameters, but may be less robust due to the limited input data and inherent difficulties associated with processing of online measurements. The model-based approaches are based on calculation of right eigenvectors of the electromechanical model of power system (see Equation (2.18)) which correspond to its dominant slow modes. These methods will be used in the thesis, as obtaining valid coherent generator groups for the sole purpose of modeling the ICI constraints is simpler and more reliable with the model-based coherency approaches. However, it should be noted that significant changes in the power system operating condition, such as topology changes or large load steps, may cause *weakly coherent* generators to change their groups [50, 133]. Therefore, online generator coherency tracking is more preferable for real implementations of ICI in physical power systems.

The well-known electromechanical power system model, which is foundational to the model-based coherency identification methods [33, 121], can be derived with the following assumptions [86]:

- The mechanical power input of synchronous generators is constant.
- Generator mechanical damping and asynchronous power are negligible.
- Synchronous generators can be represented in the network by the constant-emf-behind-the-transient-reactance model.

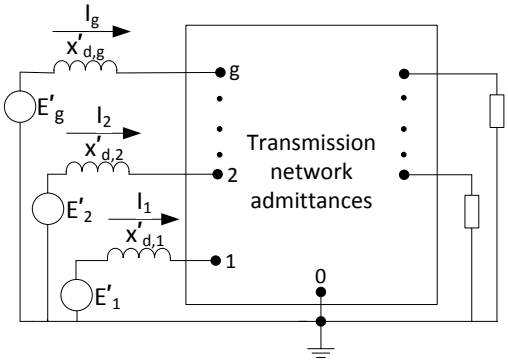


Figure 2.4: Electromechanical model of a multi-machine power system

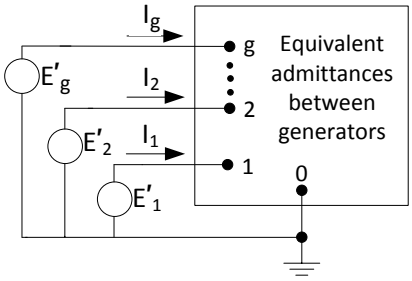


Figure 2.5: Electromechanical model reduced to internal generator nodes through Kron reduction

- The generator rotor angle coincides with the angle of the emf behind the transient reactance.
- Loads can be represented by constant impedances.

Given the assumptions above, the electric power network can be represented as shown in Figure 2.4. The network in Figure 2.4 can be reduced to contain only the internal generator buses (i.e., the nodes behind the transient reactances x'_d) by using the procedure called Kron reduction [86]. The main idea of this procedure is to eliminate the nodal equations of the original network that have zero current injections. As only the generator internal buses' nodal equations have non-zero current injections, the voltages of the remaining network nodes can be represented as a linear combination of the internal generator voltages by exploiting the fact that the left-hand sides of the network equations for the non-generator nodes are zero. The reduced network obtained through Kron reduction contains the equivalent admittances between every pair of internal generator nodes (i.e., the reduced network represents a full graph). Its schematic representation is shown in Figure 2.5.

The reduced electromechanical model can be described by (2.15), see [34, 86]:

$$\dot{\delta}_i = \omega_0(\omega_i - 1), \quad i = 1,\dots,g \tag{2.15a}$$

$$2H_i\dot{\omega}_i = P_{m,i} - \sum_{\substack{j=1\\ j\neq i}}^{g} E'_i E'_j B_{ij}\sin(\delta_i - \delta_j) - \sum_{\substack{j=1\\ j\neq i}}^{g} E'_i E'_j G_{ij}\cos(\delta_i - \delta_j)$$
$$- E'_i r_{ii}^2 - \xi_i(\omega_i - 1), \quad i = 1,\dots,g \tag{2.15b}$$

where δ_i is the rotor angle of generator i in radians, ω_i is the rotor speed of generator i in per unit, H_i is the inertia constant of generator i in seconds, ξ_i is the damping coefficient of generator i in per unit, $P_{m,i}$ is the mechanical power of generator i in per unit, E' is the constant voltage behind the transient reactance in per unit, G_{ij} and B_{ij} are the real

and imaginary components of the (i, j) entry of the admittance matrix of the reduced network (see Figure 2.5) in per unit and ω_0 is the nominal system frequency in rad s^{-1}.

The coherency behaviour of generators can be more easily understood from the linearized electromechanical model. Equations (2.15) can be linearized about an equilibrium $\delta_i = \delta_{i,0}$ and $\omega_i = 1$, where $\delta_{i,0}$ is the equilibrium rotor angle of generator i. The equilibrium rotor angles of all generators can be obtained in a convenient fashion by calculating the power flow solution for the original electromechanical model in Figure 2.5 for the loading condition of interest. The resulting linearized model derived from (2.15) is described by (2.16), see [33, 34]:

$$\Delta\dot{\delta}_i = \omega_0 \Delta\omega_i, \quad i = 1, \ldots, g \tag{2.16a}$$

$$2H_i \Delta\dot{\omega}_i = \sum_{j=1}^{g} K_{ij} \Delta\delta_j - \xi_i \Delta\omega_i, \quad i = 1, \ldots, g \tag{2.16b}$$

where $\Delta\delta_i = \delta_i - \delta_{i,0}$, $\Delta\omega_i = \omega_i - 1$ are the small perturbations of the rotor angles and speeds around their equilibrium values and K_{ij} is according to (2.17):

$$K_{ij} = E_i^{'} E_j^{'} \left(B_{ij} \cos(\delta_{i,0} - \delta_{j,0}) - G_{ij} \sin(\delta_{i,0} - \delta_{j,0}) \right), \quad j \neq i \tag{2.17a}$$

$$K_{ii} = - \sum_{\substack{j=1 \\ j \neq i}}^{g} K_{ij} \tag{2.17b}$$

Equations (2.16a)–(2.16b) can be written in the matrix form as (2.18).

$$\begin{bmatrix} \Delta\dot{\delta}_1 \\ \Delta\dot{\delta}_2 \\ \cdots \\ \Delta\dot{\delta}_g \\ \hline \Delta\dot{\omega}_1 \\ \Delta\dot{\omega}_2 \\ \cdots \\ \Delta\dot{\omega}_g \end{bmatrix} = \left[\begin{array}{cccc|cccc} & & & & \omega_0 & 0 & \cdots & 0 \\ & & & & 0 & \omega_0 & \cdots & 0 \\ & & \mathbf{0} & & \vdots & \vdots & \ddots & \vdots \\ & & & & 0 & 0 & \cdots & \omega_0 \\ \hline & & & & & & & \\ & \frac{1}{2}\mathbf{H}^{-1}\mathbf{K} & & & & -\frac{1}{2}\mathbf{H}^{-1}\Xi & & \end{array}\right] \begin{bmatrix} \Delta\delta_1 \\ \Delta\delta_2 \\ \cdots \\ \Delta\delta_g \\ \hline \Delta\omega_1 \\ \Delta\omega_2 \\ \cdots \\ \Delta\omega_g \end{bmatrix} \tag{2.18}$$

where

$$\mathbf{H} = \text{diag}(H_1, H_2, \ldots, H_g)$$
$$\Xi = \text{diag}(\xi_1, \xi_2, \ldots, \xi_g)$$
$$\mathbf{K} = [K_{ij}]$$

By expressing $\Delta\omega_i = \dfrac{\Delta\dot{\delta}_i}{\omega_0}$ and neglecting damping it is possible to reduce (2.18) to (2.19), which is the common form of the power system electromechanical model used for slow coherency analysis [33, 34]. The motivation to represent synchronous generators by simplified second order models and to neglect the damping is based on the

observation that the coherent generator groups do not depend significantly on the level of detail used in modelling the generating units [33, 55].

$$\Delta\ddot{\boldsymbol{\delta}} = \frac{1}{2}\mathbf{H}^{-1}\omega_0\mathbf{K}\Delta\boldsymbol{\delta} \tag{2.19}$$

The properties of the linearized dynamic models (2.18)–(2.19) can be analyzed by studying the eigenvalues and eigenvectors of their state matrices. In particular, the standard small-signal stability analysis [1, 134] can be performed for the model described by (2.18) in order to extract mode shapes corresponding to the electromechanical state variables $\Delta\delta$ and $\Delta\omega$. It should be noted that Equation (2.19) fully describes the properties of the complete state-space model (2.18), given that the damping is neglected in (2.18). In particular, if λ_i is an eigenvalue of the state matrix of (2.19), then $\pm\sqrt{\lambda_i}$ are the eigenvalues of the state matrix of (2.18) with all damping constants set to zero [34]. Thus, the eigenvalues of the state matrix of (2.19) correspond to the oscillatory and aperiodic modes of (2.18).

The next important question is the estimation of the number of slow inter-area modes, whose corresponding eigenvectors are used for the identification of coherent generator groups. If the number of slow modes is not given, it is common to use the *eigengap heuristic* to determine the point of separation between slow and fast electromechanical modes:

$$\chi_i = \frac{|\mathrm{Im}(\lambda_j)|}{|\mathrm{Im}(\lambda_{j+1})|}, \quad j = 2,\ldots,g-1, \quad i = 1,\ldots,g-2 \tag{2.20}$$

where λ_j is the j^{th} real eigenvalue of the state matrix of Equation (2.19), and all λ_j have been sorted in the increasing order of their magnitudes. Given such ordering of eigenvalues, their counting starts at 2, because the state matrix of (2.19) has a zero eigenvalue that does not correspond to an oscillatory mode, while only the eigengaps between the oscillatory modes are of interest. From (2.20), the number of slowest electromechanical modes to be considered for generator grouping can be expressed as follows:

$$k = \underset{i}{\mathrm{argmin}}\,\chi_i + 1 \tag{2.21}$$

where the increment of one in (2.21) can be related to the zero eigenvalue of the state matrix in (2.19) [34]. From the algebraic perspective, the minimal number of slowest modes to separate the network is two, whereby the minimal value of $\mathrm{argmin}_i\,\chi_i$ is one.

Assuming the number of slow modes is given as k, the first k eigenvectors of the state matrix of (2.19) are combined into the eigenvector matrix $\mathbf{V}$. The constant eigenvector corresponding to the smallest zero eigenvalue is usually included into $\mathbf{V}$ [33, 34, 35, 121]. Thus, the k columns of $\mathbf{V}$ represent the rotor angle mode shapes of the k slowest electromechanical modes, while the g rows of $\mathbf{V}$ represent the coordinates of the g modeled generators in the $\mathbb{R}^k$ Euclidean space. This arrangement is similar to the previously described *spectral embedding* (see Section 2.5); it is a very common machine learning model in general (i.e., g objects described by k feature vectors). Several grouping algorithms have been developed specifically for this model, including slow coherency grouping [33, 34], tight coherency grouping [35, 121] and some others.

2.7. CLUSTERING QUALITY METRICS

Many clustering algorithms used for power systems do not evaluate any clustering-specific quality metrics. Instead, clustering is accessed by observing the resulting power system performance (e.g., SVC performance with [47, 51], or accuracy of generator coherency with [34, 121]). Although this principle is clearly valid and practical, it is often hard to devise a clustering algorithm that could directly optimize the power system performance, especially if the general-purpose optimization frameworks such as MILP or metaheuristics are not an option (e.g., due to time limitations). Thus, it may be useful to introduce some intermediate quality criteria that well correlate with power system performance, but can be conveniently optimized by the means of clustering algorithms. This way of reasoning was explicitly applied in [30], where the authors defined a number of clustering quality metrics indirectly related to power system performance and combined them into a single objective function to be optimized by a genetic algorithm.

In general, a good clustering is the one that demonstrates a high intra-cluster connectivity (cluster cohesiveness) and low inter-cluster connectivity (cluster separation), which has also been mentioned in Section 1.2.2. In graph partitioning terms, separation of the cluster $\mathcal{C}$ is straightforwardly expressed as cluster cut $\text{cut}(\mathcal{C})$ (2.5). Then the overall separation of a partitioning solution can be expressed as (2.22):

$$\text{Cut}(\mathcal{C}_1,\ldots,\mathcal{C}_k) = \frac{1}{2}\sum_{j=1}^{k}\text{cut}(\mathcal{C}_j) \tag{2.22}$$

where the $\frac{1}{2}$ factor is introduced to compensate for the double inclusion of every edge running between a pair of blocks. For the similarity-based clustering and partitioning that are mostly studied in this thesis, a low Cut value clearly means that the edges running between the partitions separate highly dissimilar objects.

In practice, cluster cut (2.5) may not characterize the cluster separation well enough. For example, if the total weight of edges inside of the cluster $\mathcal{C}$ (i.e., $\text{links}(\mathcal{C},\mathcal{C})$ (2.3)) is of roughly the same value as $\text{cut}(\mathcal{C})$, the cluster cannot be considered well-separated, even if its cut is small in the absolute terms. Thus, an additional metric called cluster *expansion ratio* (or *expansion*) is introduced to highlight the expected contrast between the cluster boundary and cluster contents:

$$\phi(\mathcal{C}) = \frac{\text{cut}(\mathcal{C})}{\text{vol}(\mathcal{C})} \tag{2.23}$$

Minimizing the cluster expansion (2.23) promotes a high sum of internal connections (high volume) combined with a low sum of external connections (low cut), which are the desirable properties of a good power network partitioning according to [30] (see also Section 1.2.2). The value of $\phi(\mathcal{C})$ can take values from zero to one, with smaller values corresponding to better clusters. In addition, the previously introduced Ncut metric (2.7) can be actually understood as the arithmetic mean of cluster expansion ratios, which implies that normalized spectral clustering described in Section 2.5 is directed towards the minimization of cluster expansion ratios.

The overall partitioning quality can be accessed by the maximal expansion ratio over all clusters [76]:

$$\phi_{max}(\mathcal{C}_1,\ldots,\mathcal{C}_k) = \max_{1\le j\le k} \phi(\mathcal{C}_j) \tag{2.24}$$

Asking for a small maximal expansion ratio is reasonable from the power system point of view, because it is usually desirable in practice to avoid *any* loose clusters. A low value of (2.24) implies that all clusters are well separated from each other in terms of the similarity relationships determined by the graph adjacency matrix **W**. Thus, (2.23) and (2.24) evaluate the solution in terms of the defined power system model itself. At the same time, (2.24) allows for an efficient optimization via the normalized spectral clustering, while certain more complex objective functions can only be optimized via metaheuristic approaches (e.g., [30]).

Alternatively, the Ncut criterion (2.7) is useful to evaluate the overall partitioning quality, as it has the meaning of average expansion ratio over all clusters. The Ncut criterion can be especially useful to compare different clusterings with the same ϕ_{max} value.

According to the partitioning quality criteria introduced in Section 1.2.2, a good partitioning should also contain no disconnected or too small clusters. To account for the latter requirement, the minimal cluster cardinality (as percentage of the average cluster cardinality) is introduced as a quality indicator:

$$\varepsilon(\mathcal{C}_1,\ldots,\mathcal{C}_k) = \frac{\min_{1\le j\le k}(|\mathcal{C}_j|)}{n/k}\cdot 100 \tag{2.25}$$

Cluster sizes are treated separately from the partitioning quality measures (2.24) and (2.7). While (2.24) and (2.7) should be ideally as low as possible, (2.25) is only meant to be higher than a certain predefined minimal cluster size.

For the case of constrained network partitioning, the number of constraints violated by a clustering algorithm becomes an important issue. The most difficult type of constraints considered in this thesis are the *node grouping constraints*. As it was mentioned in Section 1.2.2, these constraints is related to the *packing Steiner trees problem*, which may even be unsatisfiable for certain combinations of the input data and constraints. To access the satisfaction of node grouping constraints, the *number of misplaced terminal nodes* is introduced by (2.26) as the number of terminal nodes not grouped into the same connected partition with the larger part of their node group. Cluster connectedness is assumed here, otherwise satisfying the node grouping constraints would become trivial.

$$\eta(\mathcal{C}_1,\ldots,\mathcal{C}_k) = \sum_{j=1}^{k} |\mathcal{T}_j \setminus \mathcal{C}_j| \tag{2.26}$$

where $\{\mathcal{T}_1,\ldots,\mathcal{T}_k\}$ is the set of node grouping constraints, and $\mathcal{C}_j$ is the connected network cluster containing the largest number of terminal nodes of $\mathcal{T}_j$.

Combined together, metrics (2.23), (2.24), (2.7), (2.25), and (2.26) enable a reliable evaluation of cluster separation, cluster size, and node grouping constraints satisfaction, which are some of the power system clustering requirements listed in Section 1.2.2. The cluster connectedness requirement does not need a continuous metric, as it is a yes-no requirement. The cluster cohesiveness requirement is not explicitly optimized,

as it is seen as complementary to cluster separation. In other words, for each number of clusters k, the set of k maximally separated clusters is requested. If the found k clusters are well separated, but some are not cohesive, the incohesive clusters are likely to feature some internal splits or bottlenecks. An incohesive cluster can be separated into a set of cohesive ones along its internal splitting boundaries, often at the expense of the overall cluster separation. Thus, the problem of balancing cluster cohesion vs. cluster separation is related to the problem of determining the optimal number of clusters.

2.8. CONCLUSIONS

This chapter laid the groundwork for the rest of the thesis. The introductory survey about SVC and ICI introduced the reader to the two main researched WAMPAC applications. The overview of graph clustering should help putting the work into perspective by describing the main classes of existing graph clustering techniques.

The remaining sections of the chapter described the important theoretic concepts. This description started by defining the mathematical notation for the thesis. Consequently, the links between normalized graph cuts and graph spectral embedding were revealed, thus providing an insight into computing several lowest eigenvectors of graph Laplacian matrices. In addition, several different spectral clustering formulations were presented and analyzed in terms of their computational efficiency. This was followed by a brief overview of generator slow coherency, including the terms and definitions, mathematical formulation, selection of the number of slow modes and similarity to spectral embedding. The final section of the chapter explained the chosen clustering quality metrics and the rationale behind them.

3

PRE- AND POST-PROCESSING FOR GRAPH PARTITIONING

3.1. INTRODUCTION

From the overview of clustering in Section 1.2, it is clear that graph partitioning is a complex yet important problem with many inherent trade-offs. Because of this, auxiliary algorithms are often used in practice to enhance the quality and flexibility of graph partitioning. For example, the obtained clustering can often be further improved by applying local search techniques as clustering post-processing [92, 135, 136]. In many situations, modifying the graph prior to its partitioning (i.e., *pre-processing* the graph) may help to improve the results or achieve specific clustering objectives [58, 59, 107, 137].

The focus of the chapter is on several prevalent shortcomings of popular graph partitioning algorithms that are often neglected in application-oriented studies. Although these shortcomings are of practical importance, they are not unique to power systems, so it is beneficial to discuss them separately from power system applications based on graph partitioning. By focusing on the issues characteristic to generic graph partitioning approaches, this chapter also serves as a gradual introduction into partitioning of power networks[1].

In this chapter, the shortcomings of graph partitioning are resolved by the use of auxiliary pre- and post-processing algorithms, which is a well-established approach in the graph partitioning community [66, 90]. The chapter describes two algorithms for graph partitioning post-processing: one to ensure cluster connectedness by systematically removing excessive minor connected components and the other to improve the cluster separation by small local changes of the partitioning cutset. For graph pre-processing, the chapter covers basic manipulations with graphs, such as edge weight modification and graph reductions. It also describes an algorithm for graph outlier detection that can be used to handle the detected outliers in advance.

[1] The material of this chapter is based on [C2, C3, C1].

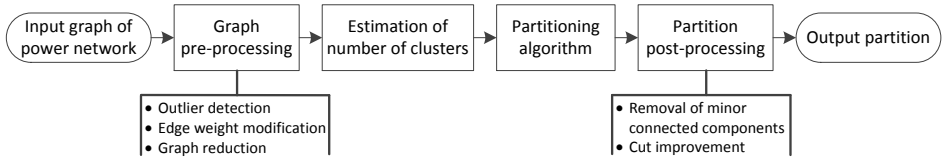


Figure 3.1: Generic graph partitioning framework

The overall representation of the graph partitioning framework relevant for this thesis is shown in Figure 3.1. Each of the elements of this framework except of the *partitioning algorithm* block is optional. In other words, each of the proposed pre-processing and post-processing algorithms can be used separately from the others to improve the specific performance aspects of various graph partitioning algorithms. The estimation of number of clusters is not considered in this chapter, but in Chapter 4.

The rest of the chapter is organized as follows. Section 3.2 discusses the graph models used in this chapter and explains the setup of the computational experiments. Section 3.3 describes the algorithm to ensure graph cluster connectivity and the graph cut improvement algorithm based on label propagation [138]. Section 3.4 explains graph modifications to ensure node grouping (*must-link*) constraints at the clustering stage and details the algorithm to detect weakly connected outliers in networks. Section 3.5 finalizes the chapter by summarizing it and drawing the conclusions.

3.2. STUDY FRAMEWORK

This section is going to describe the common aspects of the studies in Sections 3.3 and 3.4 such as the involved types of graphs and the computational setup.

3.2.1. RELEVANT GRAPHS TYPES

Depending on the goal, different properties of an electric power network can be represented through various graph models. For example, full graphs of synchronizing torque coefficients are used to model generator slow coherency, and bipartite graphs of voltage sensitivities are used for the SVC modeling purposes (see Chapter 5). The examples and studies in this chapter revolve around three types of graphs: topology graphs, series branch admittance graphs and active power flow graphs.

Topology graphs represent the network connectivity. Their set of nodes $\mathcal{V}$ represents the network buses, set of edges $\mathcal{E}$ represents the network branches (e.g., transmission lines or transformers). The edges are undirected with $(i, j) \in \mathcal{E}$ if and only if $(j, i) \in \mathcal{E}$, and their weights are given by $W_{ij} = 1, \forall (i, j) \in \mathcal{E}$.

Series branch admittance graphs share the same sets of nodes and edges with topology graphs. Their edge weights are given by $W_{ij} = 1 \Big/ \left(\sqrt{R_{ij}^2 + X_{ij}^2}\right), \forall (i, j) \in \mathcal{E}$, where R_{ij} is the series branch resistance and X_{ij} is the series branch reactance. For brevity, these graphs are going to be referred to as *branch admittance graphs*. As HV power networks usually have a low R/X ratio, the edge weights of branch admittance graphs of such networks are given by $W_{ij} = 1/X_{ij}, \forall (i, j) \in \mathcal{E}$. Admittance graphs have been used in a number of power system studies (e.g., [30, 31, 37, 67, 76, 120]) to characterize the

structure of transmission power networks and define zones for control or model reduction purposes. The use cases of branch admittance graphs are usually motivated by the need to identify strongly electrically interconnected groups of buses.

Active power flow graphs share the same sets of nodes and edges with topology graphs. Their edge weights are given by $W_{ij} = \left(|P_{ij}| + |P_{ji}|\right)/2, \forall (i,j) \in \mathcal{E}$, where P_{ij} is the active power flow from node i to j. Active power flow graphs have been actively used for ICI [43, 57, 58, 61, 139]. Due to the low R/X ratio, the transmission losses are low in HV power networks, which results in $|P_{ij}| \approx |P_{ji}|$.

As this thesis is solely concerned with HV transmission networks, the assumptions on edge weights related to this type of networks are valid for all the studies.

3.2.2. COMPUTATIONAL SETUP

As it can be seen, branch admittance graphs and active power flow graphs reflect both the discrete interconnection structure of a power network and its certain physical properties. Branch admittance graphs are more suitable for benchmarking of several algorithms on the same dataset, as the power network branch data actually defines the network and serves as an input to the power flow. Partitioning of multi-area power networks w.r.t. branch admittances often results in intuitively pleasing clusters that largely coincide with the nominal network areas.

However, a large number of test cases is often needed to validate the performance of an algorithm. The branch admittance graphs can only provide one dataset per power network to be partitioned into k clusters. By contrast, there can be infinitely many different active power flow graphs of a single power network representing various load and generation profiles. This is the main reason why active power flow graphs are mainly used in the computational experiments of this chapter. To generate a large number of test cases, many random power flows are generated for each tested number of clusters of each test network. To generate the random power flows, the power demand of each load was modeled as a uniformly distributed random variable with the range of ±50% of the nominal power.

While smaller power networks are convenient for the illustration purposes, algorithm testing is better performed on large-scale power networks. This allows to check the algorithm scalability as the input size grows and to reveal a larger variety of outcomes that mostly occur in large graphs. Therefore, the random power flows for algorithm testing are solved on the large-scale test power networks from the MATPOWER toolbox [81, 140] (e.g., *case300*, *case1354pegase*, *case2383wp*, *case2869pegase*).

3.3. GRAPH POST-PROCESSING

To motivate the need for graph partitioning post-processing, consider an example of partitioning the branch admittance graph of the IEEE 68 bus test network [141] into 7 blocks using Algorithm 2.1 with k-medoids used in place of k-means to cluster the normalized spectral embedding. The resulting partitioning shown in Figure 3.2 has been obtained using the official MATLAB implementation of k-medoids with 20 restarts and robust centroid initialization with the k-means++ algorithm [142].

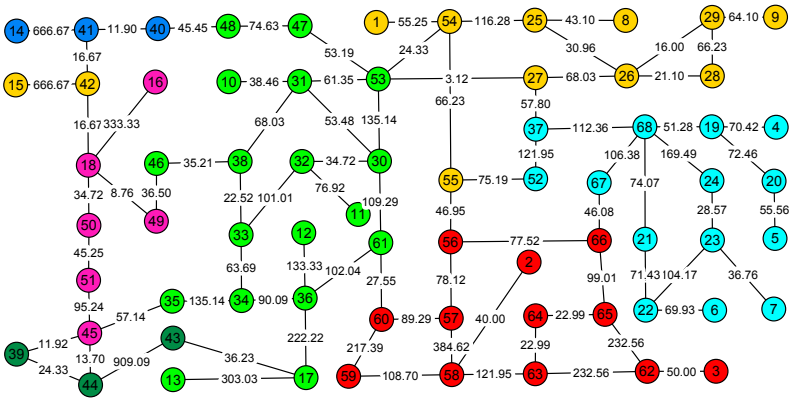


Figure 3.2: Clustering of the IEEE 68 bus test system (branch admittance graph) into 7 parts using Algorithm 2.1 with k-medoids.

From Figure 3.2 it can be immediately seen that the yellow cluster is not connected because running k-medoids on spectral embedding produced by Algorithm 2.1 does not enforce cluster connectivity. That is, algorithms for clustering point cloud data such as k-means or k-medoids aim to simply group the closest points in spectral embedding regardless of the corresponding graph nodes being connected. The k-medoids algorithm is conceptually similar to k-means, but instead of taking the mean value of data points in a cluster as the cluster center, k-medoids chooses the most centrally located data point in a cluster (i.e., the cluster medoid) as the cluster center. Compared to k-means, k-medoids has a higher computational complexity, but it is considered to be more robust to noise and outliers, which motivates its use in spectral clustering of power networks [43, 137].

Clustering spectral embedding with k-means or k-medoids may often lead to fully connected graph clusters, as spectral clustering aims to place strongly interconnected nodes close to each other in the spectral embedding. However, situations similar to Figure 3.2 are not uncommon, and their occurrence is more probable with the growth of the network size and the number of clusters. The appearance of disconnected clusters may also depend on the type of spectral embedding (e.g., (2.10), (2.11), (2.14)) and the numerical accuracy of computing eigenvectors (cf. the numerical advantages of using the matrix $\mathbf{W_n}$). The problem of cluster non-connectedness is not only restricted to Algorithm 2.1 and those similar to it, but it is a problem common to many graph partitioning algorithms (e.g., multilevel algorithms such as METIS or KaHIP [90, 92]).

In addition to the presence of non-connected clusters, several clusters in Figure 3.2 are not as well separated as they could. For example, assigning nodes 37 and 52 to the yellow cluster would produce a lower admittance cut between the yellow and cyan clusters. Thus, it is also beneficial to run graph cut improvement algorithms to increase the separation of clusters obtained after graph partitioning.

3.3.1. ENSURING CLUSTER CONNECTEDNESS

As it may be seen in Figure 3.2, the cluster non-connectedness issue is characterized by one or more clusters consisting of multiple graph connected components. Among the

connected components belonging to the same cluster, the largest one is referred to as the *major connected component*, and the remaining ones are referred to as *minor connected components*. The major connected component typically contains the vast majority of the cluster's nodes, although it is also possible to obtain minor connected components that are comparable in size with the corresponding major connected component.

To obtain a set of connected clusters, the minor connected components are reassigned to their neighboring major connected components. This process can be implemented in a number of ways, so the following requirements are introduced to guide the algorithm.

1. The reassignment of minor connected components should serve the purpose of improvement the overall clustering quality metric.

2. An assignment of a large group of nodes to a major connected component in one step should be avoided to reduce the risk of large suboptimal reassignments.

In addition to the above requirements, it is also possible to introduce a number of other useful objectives (e.g., to use the removal of minor connected components to balance cluster sizes). However, managing multiple and often conflicting objectives is not a straightforward task. Therefore, it is decided to focus only on optimizing the clustering quality. The proposed algorithm for the removal of minor connected components is a greedy heuristic inspired by the C code of a similar algorithm in METIS v5.1.0 [90] that is summarized as Algorithm 3.1.

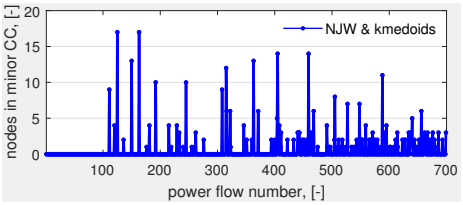
Algorithm 3.1 Removal of minor connected components

Input: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, matrix $\mathbf{W}$, clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$

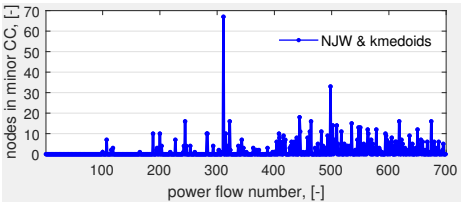
1: $\Gamma_1, \ldots, \Gamma_p \leftarrow$ connected components of $\mathcal{G}$ induced by $\mathcal{C}_1, \ldots, \mathcal{C}_k$.
2: **if** $p = k$ **then return** $\Gamma_1, \ldots, \Gamma_k$ **end if**
3: Reorder $\Gamma_1, \ldots, \Gamma_p$ in the decreasing order of their number of nodes.
4: Define $\Gamma_1, \ldots, \Gamma_k$ as the major connected components.
5: Define $\Gamma_{k+1}, \ldots, \Gamma_p$ as the minor connected components.
6: $\Gamma_{k+1}, \ldots, \Gamma_q \leftarrow$ subdivide $\Gamma_{k+1}, \ldots, \Gamma_p$ that are larger than 10% of the size of Γ_k.
7: **for** $i \leftarrow 1, \ldots, q-k$ **do**
8: $\quad \Gamma_j \leftarrow$ the minor connected component with the highest total connection weight to a single major connected component.
9: $\quad$ Sequentially add Γ_j to all adjacent major connected components.
10: $\quad$ Finally implement the assignment of Γ_j that reduces ϕ_{max} (or Ncut for equal changes of ϕ_{max}) the most.
11: **end for**

Output: $\Gamma_1, \ldots, \Gamma_k$ // Major connected components after assigning to them all minor ones

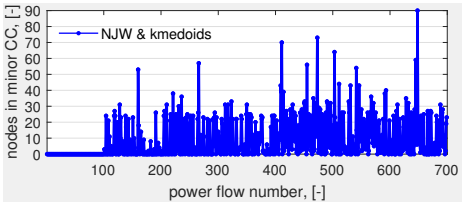
In step 6 of Algorithm 3.1, the number of minor connected components is increased if some of them are comparable in size with the smallest major connected component. Such large minor connected components are recursively bi-partitioned into smaller ones until there are no minor connected components exceeding the predefined size threshold. This size threshold is set to 10% of the number of nodes in the smallest major



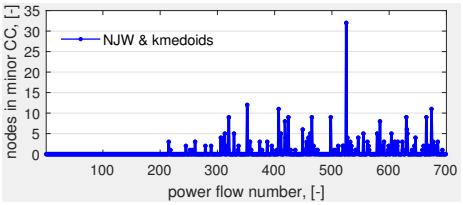
(a) case300 test network



(b) case1354pegase test network



(c) case2383wp test network



(d) case2869pegase test network

Figure 3.3: Total number of nodes in minor connected components after partitioning of random power flow graphs of the 4 power networks from MATPOWER with 300, 1354, 2383, and 2869 nodes using Algorithm 2.1

connected component; it can also be set lower than 10%, up to deciding on each node individually, but this would increase the total number of iterations of Algorithm 3.1. The used bi-partitioning algorithm is recursive bisection of the minor connected component by the second eigenvector of its Laplacian matrix (2.6a), also known as Fiedler vector [105]. Recursive bisection using graph min-cut algorithms [125] could be appropriate as well. In Algorithm 3.1, the maximal expansion ratio over all clusters (2.24) is chosen as the minimization metric with the goal to produce an improved partitioning in which no single cluster is significantly worse than the others. Often the current minor connected component has no connection to the least balanced partition, which explains the minimization of the total normalized cut (2.7) chosen as the secondary objective.

To highlight the use cases of Algorithm 3.1, 700 randomly generated active power flow graphs of the 4 large-scale networks from MATPOWER (*case300*, *case1354pegase*, *case2383wp*, *case2869pegase*) have been partitioned into 2,...,8 clusters using Algorithm 2.1 with k-medoids (20 restarts and k-means++ initialization). Each number of clusters has received 100 randomly generated power flow cases that were generated as described in Section 3.2.2. From Figure 3.3, it can be seen that the proposed Algorithm 3.1 needs to reassign only a small fraction of network nodes, while the larger part becomes initially assigned to the major connected components.

3.3.2. GRAPH CUT IMPROVEMENT

The great strength of spectral graph partitioning methods lies in their ability to extract the global graph connectivity from the first k eigenvectors of graph Laplacian matrices. However, finding clusters that reflect the global picture of the graph structure still often results in suboptimal clusterings, in which the boundary nodes of some clusters could

be moved to the neighboring clusters to increase the cluster separation (measured as (2.5) or (2.7)). The property of spectral clustering and many other graph partitioning algorithms to compute suboptimal clusterings is well-known, and a common remedy for this issue is to apply a heuristic improvement algorithm to the obtained partitioning solution.

A label propagation based greedy local search as described in [138] is a simple and time-efficient algorithm that can be used to refine the solution obtained by spectral clustering. Given the initial partitioning boundary, the algorithm reassigns the boundary nodes to different clusters if it reduces the total graph cut and proceeds until no more improvement can be found. The idea of this graph cut improvement algorithm is shown as Algorithm 3.2.

Algorithm 3.2 Label propagation based graph cut improvement

Input: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, matrix $\mathbf{W}$, clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$, iteration limit t_{max}
for $j \leftarrow 1, \ldots, k$ **do**
  $\mathcal{C}_j^0 \leftarrow \mathcal{C}_j$ // Initialize cluster node sets at step 0 as $\mathcal{C}_1^0, \ldots, \mathcal{C}_k^0$
end for
for $t \leftarrow 1, \ldots, t_{max}$ **do**
  for $v_i \in \mathcal{V}$ **do**
    $j^* \leftarrow \text{argmax}_j \, \text{links}(v_i, \mathcal{C}_j)$ // (2.3)
    $\mathcal{C}_{j^*}^t \leftarrow \mathcal{C}_{j^*}^{t-1} \cup \{v_i\}$
  end for
  if $\mathcal{C}_1^t, \ldots, \mathcal{C}_k^t = \mathcal{C}_1^{t-1}, \ldots, \mathcal{C}_k^{t-1}$ **then break end if**
end for
Output: $\mathcal{C}_1^t, \ldots, \mathcal{C}_k^t$ // Final clusters after all node swaps

The used version of the label propagation algorithm [143] is based on the KaHIP multilevel graph partitioning library [92], which was compiled in a dedicated way to produce a stand-alone executable for the label propagation based graph cut improvement. It is common for multilevel graph partitioners to ensure that no single cluster is too big, so the used implementation of Algorithm 3.2 will normally try to balance cluster sizes by avoiding *too large* clusters. When partitioning power networks, it is rather common to balance cluster sizes by avoiding *too small* clusters. The source code of KaHIP could be modified to change the native cluster size balance constraint to the power system specific one, but it was decided not to implement this feature and to completely relax the built-in cluster size constraint instead.

As it can be seen, Algorithm 3.2 focuses on improving the cluster separation by minimizing the graph cut through maximizing the internal connectedness of each cluster. Because the sum of all edge weights is fixed, maximizing the sum of intra-cluster edge weights is equivalent to minimizing the sum of inter-cluster edge weights [67, 122]. Refining the graph cut often simultaneously reduces the normalized cut (2.7) and maximal expansion ratio (2.24), except the situations when minimizing the graph cut too much reduces the volume of some clusters. In addition, minimizing the active power flow cut is relevant for ICI [43, 57]. Therefore, local search algorithms for graph cut improvement (e.g., Algorithm 3.2) can be seen as a versatile tool to refine various power network areas.

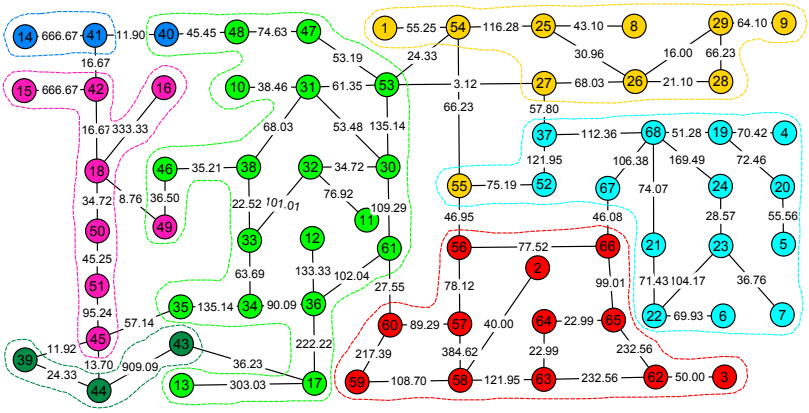
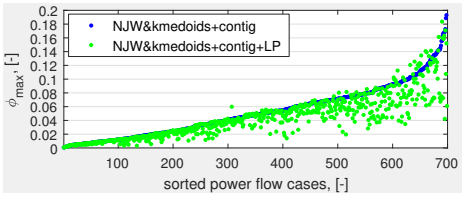


Figure 3.4: Results of applying the clustering connectedness Algorithm 3.1 and the graph cut improvement Algorithm 3.2 to the partitioning in Figure 3.2. The node colors illustrate the outcome of Algorithm 3.1, and the dashed contours show the clusters' boundaries after the subsequent run of Algorithm 3.2.

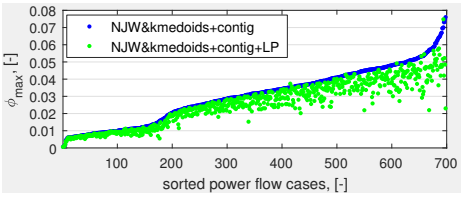
An illustration of the effect of the greedy local search Algorithm 3.2 on the area separation is shown in Figure 3.4. As it can be seen, Algorithm 3.2 changes the cluster assignments of a small fraction of the network nodes (namely, nodes 40, 49, and 55 change their clusters). These small changes are well-justified, and their impact is noticeable too: the total admittance cut drops from 498.64 p.u. to 428.39 p.u., the normalized cut (2.7) drops from 0.0644 to 0.0554, and the maximal expansion ratio (2.24), which corresponds to the yellow partition, drops from 0.1593 to 0.1360.

To further illustrate the effects of the label propagation based graph cut improvement on the partitioning quality, Algorithms 3.1 and 3.2 are applied to the same set of random power flow scenarios that was used in Section 3.3.1. At first, the possible minor connected components are removed by Algorithm 3.1, and then the resulting partitioning is improved with Algorithm 3.2. The results of the graph cut improvement stage are shown in Figures 3.5, 3.6, 3.7. To clearer show the impact of Algorithm 3.2, the order of test cases in Figures 3.5, 3.6, 3.7 is adjusted in the increasing order of the given parameter (i.e., ϕ_{max}, Ncut, or ε) *before* Algorithm 3.2. This creates a clearly recognizable monotonically increasing sequence of the baseline results, which otherwise would be non-monotonic and highly scattered due to the high randomness of the power flow generation process.

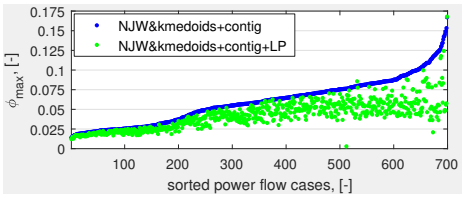
As it can be seen from Figures 3.5, 3.6, 3.7, the graph cut minimization with Algorithm 3.2 also decreases the Ncut and ϕ_{max} metrics in the majority of cases, and the improvement is often very significant. Moreover, despite of not including any cluster size constraints, the minimal cluster size tends to vary little due to Algorithm 3.2. It has been observed that Algorithm 3.2 typically reassigns only a small fraction of nodes to different clusters. Here it is worth to recall that the constraints on the minimal cluster size could be seamlessly added to Algorithm 3.2, which would further improve its cluster size balancing properties (this step was skipped to save the implementation effort and to show the worst-case scenario when no cluster size constraints are present).



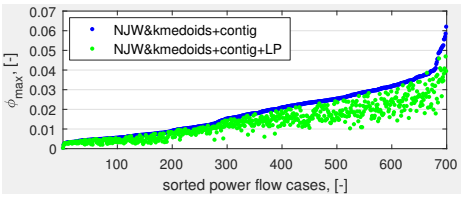
(a) case300 test network



(b) case1354pegase test network



(c) case2383wp test network



(d) case2869pegase test network

Figure 3.5: Maximal expansion among clusters after partitioning of random power flow graphs of the 4 power networks from MATPOWER with 300, 1354, 2383, and 2869 nodes using Algorithm 2.1 and running Algorithm 3.1 (blue), with the subsequent graph cut improvement using Algorithm 3.2 (green).

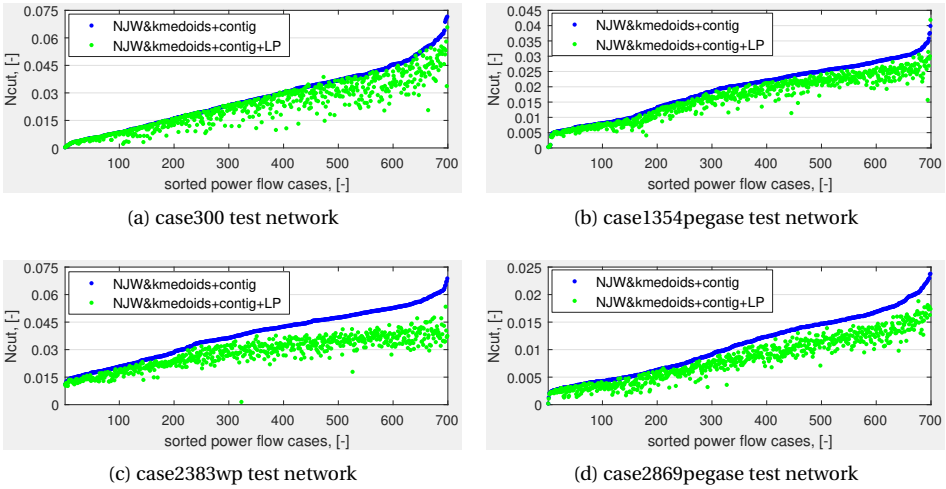
3.3.3. COMPUTATIONAL TIME OF POST-PROCESSING STEPS

To give an idea about the computational burden of Algorithms 3.1 and 3.2, the run times of various clustering steps measured by the *timeit()* routine of MATLAB are summarized in Table 3.1. In the table header, T_{SpCl} is the time of Algorithm 2.1 with the k-medoids clustering, T_{conn} is the time of the cluster connectedness Algorithm 3.1, N_{mcc} is the number of reassigned nodes in minor connected components, T_{LP} is the time of the graph cut improvement Algorithm 3.2. For each entry, the number of clusters was taken from 7 to 8 in order to obtain a more conservative timing estimate. The results were obtained on MATLAB R2017a (64-bit) on a PC with an Intel® Xeon® E5 3.70 GHz CPU on a single core using a Linux virtual machine with 2 Gb of RAM.

n	T_{SpCl}, [ms]	N_{mcc}	T_{conn}, [ms]	T_{LP}, [ms]
300	150	6	34	50
1354	2300	71	470	105
2383	7100	5	52	350
2869	10500	8	64	410

Table 3.1: Computational time of post-processing steps

The results clearly show that the proposed improvements have a reasonable computational cost and scale well with the increasing number of buses. It should be noted that the time T_{SpCl} is dominated by the k-medoids algorithm. Using k-means instead would decrease the computational time significantly, at some expense in robustness to outliers. The run time of the connectedness heuristic mostly depends on the actual number of buses belonging to minor connected components and the actual composition of these



(a) case300 test network

(b) case1354pegase test network

(c) case2383wp test network

(d) case2869pegase test network

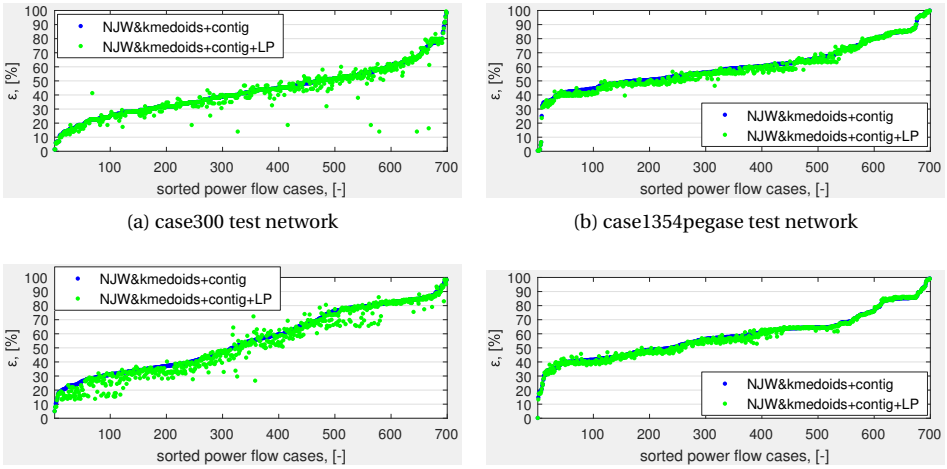
Figure 3.6: Normalized cut after partitioning of random power flow graphs of the 4 power networks from MATPOWER with 300, 1354, 2383, and 2869 nodes using Algorithm 2.1 and running Algorithm 3.1 (blue), with the subsequent graph cut improvement using Algorithm 3.2 (green).

components and thus can vary significantly. The run time of Algorithm 3.1 could also be lower if it was implemented in a high-performance programming language instead of the used prototyped implementation in MATLAB. The run time of the label propagation based graph cut improvement is consistently low due to its efficient implementation in the C++ language provided by the KaHIP graph partitioning library [92, 138]. The asymptotic time complexity of the label propagation algorithm is proportional to the number of graph edges m, but hard to prove mathematically [143] due to uncertainty about the total number of required iterations.

3.4. GRAPH PRE-PROCESSING

Graph pre-processing is a popular approach in many domains to improve the speed and quality of graph clustering algorithms. For example, graph edge weight modification can be used if the edge weights lie in a narrow range of values (e.g., Pearson correlation coefficients [68]) or vary too greatly (logarithmic or square root transformations may be used in this case). Many graph-based algorithms are more efficient on sparse graphs with the number of edges being much lower that the square of the number of nodes [68, 144]. In addition, sparse graphs require less storage space and are easier to visualize. These advantages motivate graph sparsification as another graph pre-processing tool. Other relevant graph pre-processing tasks include search for hub and outlier nodes [145, 146], symmetrization of directed graphs [147], specialized graph transformations to improve clustering (e.g., b-matchings [148]), graph coarsening [90] etc.

The relevance of each pre-processing task varies a lot depending on the target application. In the case of electric power networks, graph sparsity is often inherently present, as the number of transmission links is kept low due to their high construction costs.



(a) case300 test network

(b) case1354pegase test network

(c) case2383wp test network

(d) case2869pegase test network

Figure 3.7: Relative minimal cluster size after partitioning of random power flow graphs of the 4 power networks from MATPOWER with 300, 1354, 2383, and 2869 nodes using Algorithm 2.1 and running Algorithm 3.1 (blue), with the subsequent graph cut improvement using Algorithm 3.2 (green).

The edge weights are typically meaningful physical parameters (e.g., power flows, admittances, impedances, sensitivities etc.), which are kept in their original form without applying edge weight transformations.

This section focuses on the two tasks that were important for the developments of this thesis and occurred consistently in the studies. The first part of the section introduces the use of graph reductions to enforce various power system specific node grouping constraints at the clustering stage. The second part of the section sheds light on the quick detection of connectivity outliers in similarity graphs of large-scale power networks. The problem of outliers manifests itself in Figure 3.7, which highlights the insufficiency of the robustness properties of k-medoids to fully prevent very small clusters. Namely, Figure 3.7 demonstrates that a number of clusterings has very low minimal cluster sizes. The proposed outlier detection algorithm can be used for graph pre-processing as well as a stand-alone analytic method.

3.4.1. GRAPH REDUCTIONS FOR CONSTRAINED GRAPH PARTITIONING

By adopting the traditional machine learning terminology, the node grouping constraints introduced in Section 1.1.4 can be characterized in terms of *must-link* and *cannot-link* constraints. Must-link implies that the nodes must be in the same partition, and cannot-link implies the nodes cannot be in the same partition.

From the power system perspective, the constrained graph partitioning requirements relevant for this thesis have been listed in Section 2.2.2 as constraints 1–4. The generator coherency node grouping constraint includes both must-link (each coherent group must belong to a single partition) and cannot-link (different coherent groups cannot be in the same partition) constraints. The transmission line availability constraint

can be viewed as a must-link constraint for the terminals of the constrained network branch. The blackstart unit availability constraint is similar to the generator coherency constraint but confined to generators with blackstart capability.

Must-link constraints stemming from the transmission line availability constraints are uniquely defined (i.e., a strictly defined subset of edges $\mathcal{E}^{ML} \subset \mathcal{E}$ that cannot be disconnected). In contrast, must-link constraints stemming from the generator coherency constraints can often be satisfied in a number of ways, as the generator coherency requirement only specifies the sets of nodes to be kept together (i.e., a node grouping constraint), without specifying the exact edges to interconnect each group of nodes.

A convenient way to satisfy the transmission line availability constraints is through contraction of must-link edges [58, 149]. Once an edge is contracted to a single node, it cannot be disconnected, and the both edge terminals are guaranteed to belong to the same partition. Due to their exact definition, transmission line availability constraints can be ensured independently and prior to the general node grouping constraints such as generator coherency constraints. However, the consistency of constraints should be verified first: there should be no must-link edge connecting two generator nodes belonging to different coherent generator groups.

In [150], the authors proposed an efficient method that can be used to aggregate the must-link constraints stemming from an exact set of edges $\mathcal{E}^{ML}$, which is summarized here as Algorithm 3.3.

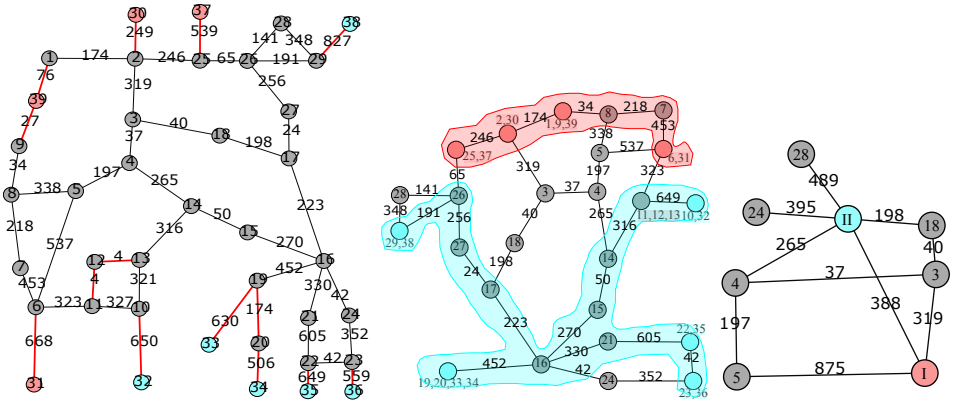
Algorithm 3.3 Encoding of branch unavailability constraints through graph reduction

Input: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mu, W)$, set of unavailable edges $\mathcal{E}^{ML}$.

1: Create a simple unweighted graph $\mathcal{G}^{ML} = (\mathcal{V}, \mathcal{E}^{ML})$ including all unavailable edges.
2: $\Gamma_1^{ML}, \dots \Gamma_r^{ML} \leftarrow$ connected components of $\mathcal{G}^{ML}$ larger than one node.
3: **for** $j \leftarrow 1, \dots, r$ **do**
4: Merge the nodes of $\mathcal{G}$ belonging to Γ_j^{ML} into node p. Node p inherits the group of any nodes in Γ_j^{ML} that are subject to general node grouping constraints.
5: $\mu(p) \leftarrow \sum_{v_i \in \Gamma_j^{ML}} \mu(v_i)$
6: Add an edge e_{pq} between the merged nodes $v_i \in \Gamma_j^{ML}$ and any other node v_q in $\mathcal{G}$ if there are edges between $v_i \in \Gamma_j^{ML}$ and v_q. Set $W(e_{pq}) \leftarrow \sum_{v_i \in \Gamma_j^{ML}} W_{iq}$.
7: $\mathcal{E} \leftarrow \mathcal{E} \setminus \left\{ e_{ij} | v_i \in \Gamma_j^{ML} \text{ or } v_j \in \Gamma_j^{ML} \right\}$
8: $\mathcal{V} \leftarrow \mathcal{V} \setminus \Gamma_j^{ML}$.
9: **end for**

Output: Reduced graph $\mathcal{G}' = (\mathcal{V}, \mathcal{E}, \mu, W)$.

In Algorithm 3.3, the graph node weights depend on the application. For normalized spectral clustering, the node weights are set to the weighted node degrees (see Section 2.5.1). Setting the degree of the aggregate node to the sum of degrees of the merged nodes at line 5 of Algorithm 3.3 ensures the equivalence of all normalized cuts that *do not include the unavailable branches* in the initial and reduced graphs, with the proof being given in [150]. In other words, partitioning of the reduced graph $\mathcal{G}'$ always satisfies the branch unavailability constraints, and the resulting normalized cuts (2.7) (or regular



(a) Power flow graph of the IEEE 39 bus system

(b) Unavailable branches merged, generator group trees highlighted

(c) Generator group trees merged

Figure 3.8: Graph reduction process of the active power flow graph of the IEEE 39 bus test case. The edge weights represent the active power flows in MW. The generator groups are indicated with cyan and red node colors. In addition to all transformers, lines (1,39) and (9,39) are constrained as unavailable branches and colored in red.

graph cuts (2.22), or any other graph cuts based on cut and volume) preserve their value when mapped back to the original graph $\mathcal{G}$.

The main idea of Algorithm 3.3 lies in merging groups of nodes that are found as connected components of graph $\mathcal{G}^{ML}$. These connected components are linked by unavailable edges, which may be contracted to produce a reduced graph that is guaranteed to be connected. General node grouping constraints can also be enforced through node collapse, but the set of edges that should be contracted is not given in this case. As it has been discussed in Section 1.2.2, selecting the k sets of edges to connect the k groups of terminal nodes is an NP-complete problem known as the Steiner tree packing problem [77]. This Steiner problem has been approximated through constructing k minimal spanning trees with subsequent tree trimming to obtain k Steiner trees [108] connecting each group of terminal nodes (e.g., see [58, 149]). Once the edges connecting each group of terminal nodes have been obtained, they can be contracted using the for-loop of Algorithm 3.3, thus ensuring the must-link condition on the node grouping constraints.

The steps of the graph reduction process described above are shown in Figure 3.8 on the example of the active power flow graph of the IEEE 39 bus test network. In Figure 3.8a, the unavailable branches are shown in red, and the nodes belonging to the two generator groups are colored in cyan and red. The unavailable branches include all transformer branches as well as lines (1,39) and (9,39), which are constrained in a deliberate manner to prevent possible solutions disconnecting the generator at bus 39. The generator groups in Figure 3.8a are not meant to represent the generator coherency, but rather chosen freely as an example. In Figure 3.8b, the connected components formed by the unavailable branches (e.g., {19, 20, 33, 34}) are collapsed to single nodes. In the resulting reduced network, two Steiner trees connecting the nodes of the two generator groups are highlighted. These trees can be collapsed in an analogous way to the con-

nected components formed by the unavailable branches to form the final reduced graph shown in Figure 3.8c. Once each generator group is collapsed to a node, the cannot-link coherency constraints can be ensured by assigning the remaining nodes (i.e., {3, 4, 5, 18, 24, 28} in Figure 3.8c) to the nearest generator group node. This process was illustrated in [C1] using graph distances induced by spectral embedding.

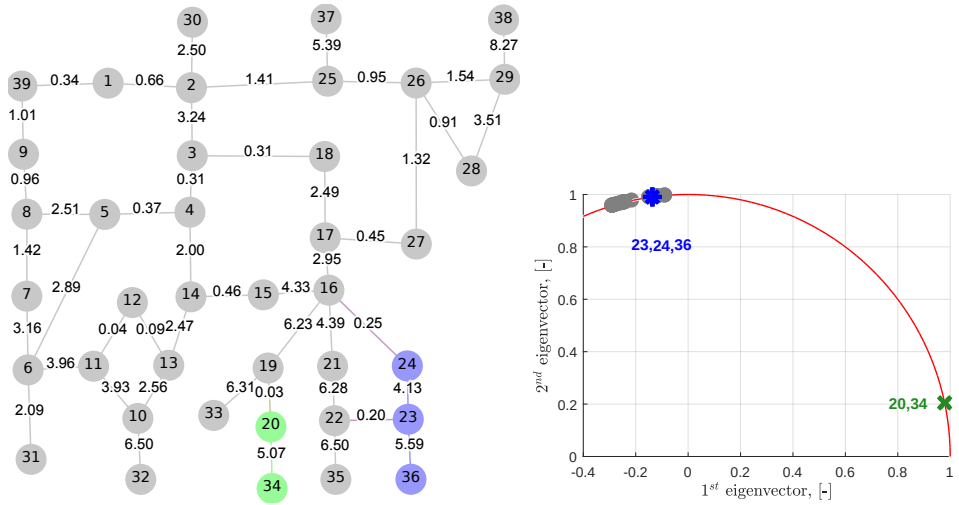
In practice, it is recommendable to perform the contraction of unavailable branches whenever the considered application benefits from it, as it allows to reduce the problem size and the number of constraints. The tree collapse procedure for general node grouping constraints is not uniquely defined, and its multiple possible solutions may vary a lot in their quality. In addition, tree construction is a sequential process; by constructing the current group tree, it is possible to discard all possible configurations of the next group tree. Thus, the sequential tree construction process [58, 149] may require multiple restarts to vary the order of construction of the group trees. Nevertheless, these drawbacks can be acceptable for an offline planning environment.

3.4.2. GRAPH OUTLIER MINING

Outliers represent a common problem in clustering of many types of data. The presence of outliers may lead the clustering algorithm to a highly unbalanced solution, or to lead the algorithm away from the optimal clustering [151]. These phenomena were demonstrated for the partitioning of electric power networks in [137]. In particular, spectral clustering is known to be sensitive to outliers [152]. This problem can be tackled by applying a robust clustering algorithm to cluster the spectral embedding [137] or by removing the outliers in advance [151]. The first option limits the set of methods to produce the final partitioning, and severe outliers may still be able to impair the partitioning result. For example, using Algorithm 2.1 with k-medoids could not prevent several very unbalanced partitionings in Figure 3.7. Thus, it is strongly recommended to filter out the severe outliers before applying spectral clustering [152]. Besides robust clustering, many applications (e.g., event detection, network anomaly detection) are directly formulated as outlier detection problems [153, E4].

Compared to outlier detection in Euclidean point cloud data, there are noticeably less methods to find outliers in graphs [154]. The majority of existing graph outlier detection methods focuses either on community outliers [145, 146] (i.e., residual nodes loosely coupled to the identified network communities), or on identifying anomalous elements in graph matrices via matrix factorization techniques [155]. These methods are not tailored to find *outlier clusters* (i.e., loosely connected groups of two or more nodes). Additionally, many state-of-the-art outlier detection methods have a higher time complexity than spectral clustering [155, 156], which may result in a computational bottleneck.

The above considerations motivate the graph outlier mining algorithm in this section. The spectral embedding of the analyzed similarity graph lies at the basis of the algorithm, which explains its scalability that is comparable to the scalability of Algorithm 2.1. The proposed algorithm detects outlier clusters by analyzing the minimum spanning tree (MST) of the spectrally embedded graph, which was introduced in Section 2.5.2. Such MST was previously used in [157] for the clustering purposes. However, an output of graph clustering may not reveal all of the outliers as they may mask each other. That is, some outliers may look similar to the normal data in the presence of a



(a) Randomly generated active power flow graph of the IEEE 39 bus test system. The edge weights are the active power flows in p.u. on the 100 MVA base.

(b) Row-normalized spectral embedding of the graph in Figure 3.9a. The 2D unit sphere is shown in red.

Figure 3.9: Graph outlier masking

substantially more severe outlier [151, 154]. For this reason, the proposed spectral minimum spanning tree (SpMST) based outlier mining searches the entire graph for loosely connected nodes and clusters below certain size to return a collection of such nodes and clusters together with their severity ranking.

An example of outlier masking is shown in Figure 3.9 that represents a randomly generated active power flow graph of the IEEE 39 bus test system from MATPOWER [81] and its row-normalized spectral embedding using the two largest eigenvectors of the matrix $\mathbf{W_n}$. In Figure 3.9a, the node groups {20, 34} and {23, 24, 36} have a relatively weak connection to the rest of the network. However, in the spectral embedding in Figure 3.9b, only the cluster {20, 34} manifests itself as an outlier, and the cluster {23, 24, 36} is masked by its presence. Thus, running an outlier detection algorithm for Euclidean point clouds (e.g., [156]) would not necessarily detect the cluster {23, 24, 36}, as it would become more visible only after the removal of the cluster {20, 34}. As the network scale increases, situations similar to Figure 3.9 become more common.

OUTLIER RANKING

It is often not practical to draw an exact border between normal data and outliers. For this reason, many outlier detection methods operate by assigning outlier scores to the data instead of directly labeling outliers. With the outlier scores available, it is possible to remove outliers starting from the most severe ones up to a user-defined threshold.

The found outlier clusters are ranked according to their expansion ratios (2.23), with the weakly-connected clusters having a low value of this parameter. Meila and Shi [158] introduced the probabilistic interpretation of cluster expansion: $\phi(\mathcal{C})$ describes the probability of the random walk defined by the transition probability matrix $\mathbf{P}$ (2.13a) of

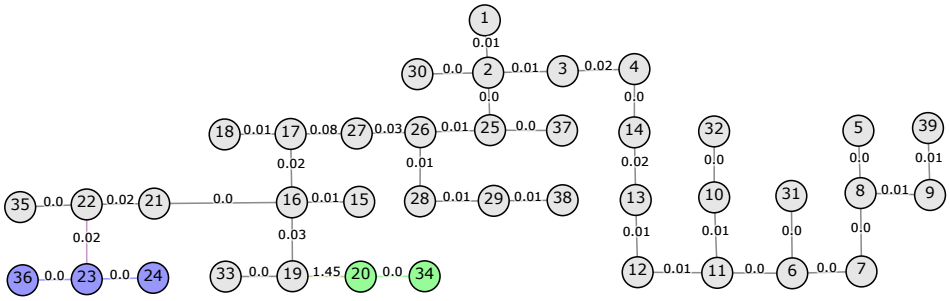


Figure 3.10: SpMST of the spectrally embedded graph obtained from the spectral embedding in Figure 3.9b, rooted at node 1.

the Markov chain associated with the graph $\mathcal{G}$ to leave cluster $\mathcal{C}$ in one step starting from the random walk stationary distribution, if the current walk state is in $\mathcal{C}$.

It is problematic to extend (2.23) to a single node cluster, as the corresponding score value will be one even if the node is very weakly connected to the rest of the network. This is consistent with the probabilistic interpretation, since, starting at any node, random walk leaves it on the next step with the probability one. From the random walk perspective, a single node outlier in a weighted undirected graph can be understood as a node that is unlikely to be visited from its neighbors. The corresponding score for node v_i can be introduced as

$$p_{max}(v_i) = \max_{\{j|p_{ij}\neq 0\}} p_{ji} = \max_{\{j|W_{ij}\neq 0\}} \frac{W_{ij}}{D_j} \tag{3.1}$$

where p_{ij} are the elements of the transition probability matrix $\mathbf{P}$ (2.13a). Score (3.1) can be efficiently evaluated for every graph node, and it attains a low value for single graph nodes that are weakly connected to any of their neighbors.

SPMST CONSTRUCTION

A SpMST can be initially constructed by a standard algorithm, such as Prim or Kruskal [159]. After the SpMST has been obtained, some preparations should be made to efficiently examine it [160].

First, one graph node is selected as the tree root, and the tree is traversed from the root with the breadth-first search (BFS) algorithm [159]. The BFS algorithm is normally capable of returning the predecessor vector, the n entries of which contain the indices of parent nodes for each node. For example, the fifth and ninth entries of the predecessor vector of the SpMST shown in Figure 3.10 should both contain the number 8. The predecessor vector is used to orient the SpMST edges from parent nodes to their successors. That is, if the SpMST edges are stored as 2-tuples, the first tuple entry should be the index of the parent (or *from*) node. Secondly, the standard BFS algorithm was modified to additionally return the direct children of each node. By its principle, the BFS algorithm starting at the top of the tree hierarchy will not leave the current node before visiting all of its children. Therefore, additionally returning a key-value map from each node to the list of its direct children is straightforward with BFS. Next, another map is created

Algorithm 3.4 Obtain descendants of each SpMST node

Input: Map from nodes to their direct children `child`, tree root node $root$.
Output: Map from nodes to all their descendants `desc`.

```
1: for i ← 1,...,n do
2:     desc{i} ← {v_i} // Initialization
3: end for
4: desc ← GETDESC(root, desc, child)
5: return desc

6: function GETDESC(node, desc, child)
7:     if child[node] = ∅ then return
8:     else
9:         for k ← child{node} do
10:            desc ← GETDESC(k, desc, child)
11:            desc{node} ← desc{node} ∪ desc{k}
12:        end for
13:    end if
14: end function
```


to store all the descendant nodes of each node in the tree. This key-value map can be created effectively by recursion as shown in Algorithm 3.4. With this map, it is easy to get the number of nodes below each node (including the node itself) as an extra vector of n elements.

OUTLIER MINING WITH FUNDAMENTAL CUTSETS

Since any spanning tree (including MST) contains the minimal number of edges to interconnect the nodes of the graph, removing an edge from it creates two connected components. The set of edges that is needed to produce the same connected components in the initial graph is called *fundamental cutset*. Since any spanning tree interconnects the n graph nodes with $n-1$ edges, a spanning tree defines $n-1$ fundamental cutsets in the graph [108]. A classical spanning tree clustering algorithm (e.g., [99, 157, 161]) would disconnect the $k-1$ longest edges of the MST to produce k clusters, thus using the $k-1$ fundamental cutsets induced by those MST edges. Since the goal of this section is not to partition the graph into k clusters, but to detect loosely connected subgraphs below certain size, all $n-1$ fundamental cutsets of the SpMST are going to be examined, which is formulated as the graph outlier mining Algorithm 3.5.

The SpMST is provided to Algorithm 3.5 as a priority queue containing the directed edges included into SpMST in the decreasing order of their lengths. Due to the availability of precomputed descendant nodes for any node, the cardinalities of connected components of each fundamental cut can be found in constant time. Then the algorithm simply checks if one of the components satisfies the cluster size criterion and, if it does, the algorithm further computes its outlier score. As the sought clusters are small, the set difference operation will only be computed a few times to return the limited number of valid outlier candidates containing the SpMST root node. Components that are smaller

than the predefined cardinality $CC_{\max}$ and posses a low enough outlier score are saved as outlier clusters.

Algorithm 3.5 Outlier mining with fundamental cutsets

Input: Priority queue `SpMST`, set $\mathcal{V}$, nodes descendants map `desc`, number of descendants of each node $\boldsymbol{nd}$, matrix $\mathbf{W}$, matrix $\mathbf{D}$, limit on cluster cardinality $CC_{\max}$, limit on outlier score $\phi_{\max}$.

$\mathcal{S} \leftarrow \emptyset$ // Set of outlier clusters
for $k \leftarrow 1, \ldots, n-1$ **do**
  $(f_k, t_k) \leftarrow$ `SpMST.pop()` // Pop current longest SpMST edge
  if $(\boldsymbol{nd}[t_k] < CC_{\max})$ **or** $(n - \boldsymbol{nd}[t_k] < CC_{\max})$ **then**
    if $\boldsymbol{nd}[t_k] < CC_{\max}$ **then**
      $\mathcal{C} \leftarrow$ `desc`$\{t_k\}$ // cluster nodes
      $CC \leftarrow \boldsymbol{nd}[t_k]$ // cluster cardinality
    else
      $\mathcal{C} \leftarrow \mathcal{V} \setminus$ `desc`$\{t_k\}$
      $CC \leftarrow n - \boldsymbol{nd}[t_k]$
    end if
    $\mathrm{vol}(C) \leftarrow \sum_{v_i \in \mathcal{C}} D_i$ // (2.4)
    $\phi_{\mathrm{LB}} \leftarrow \mathbf{W}[f_k, t_k] / \mathrm{vol}(\mathcal{C})$ // Lower bound on ϕ
    if $\phi_{\mathrm{LB}} < \phi_{\max}$ **then**
      if $\phi(\mathcal{C}) < \phi_{\max}$ **then** $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{C}$ // score $\phi(\mathcal{C})$ is also stored
      end if
    end if
  end if
end for

Output: $\mathcal{S}$, outlier scores.

OUTLIER MINING WITH SINGLE-LINK CLUSTERING

Outlier mining with fundamental cutsets is based on *divisive clustering* of MST, as each iteration of Algorithm 3.5 divides the SpMST into two parts and evaluates them. However, some clusters with relatively high scores (2.23) can only be discovered by removing more than one long edge in the SpMST. This may also depend on the number of used eigenvectors (e.g., a cluster may become detectable with Algorithm 3.5 if more eigenvectors of $\mathbf{W_n}$ are considered). Nevertheless, introducing an additional bottom up (or *agglomerative*) clustering mechanism makes the outlier mining more robust in general.

Agglomerative outlier mining uses the known analogy between single-link hierarchical clustering and the Kruskal algorithm for MST [161]. The Kruskal algorithm is a classical MST algorithm that runs in $O(|\mathcal{E}|\log|\mathcal{E}|)$ time by using the union-find data structure (as given in [159]). The relevant ideas of the Kruskal algorithm are included into the agglomerative outlier mining Algorithm 3.6 to provide a clear and complete presentation.

As Algorithm 3.6 operates on the already constructed MST without any loops, it does not need to check if the two nodes are already in the same connected component. In-

stead, the cardinality of the resulting connected component is checked, and if a too large component results from adding the next edge, this edge is skipped.

Algorithm 3.6 Outlier mining with single-link clustering

Input: Priority queue SpMST, matrix **W**, matrix **D**, limit on cluster cardinality $CC_{\max}$, limit on outlier score $\phi_{\max}$.

$\mathcal{S} \leftarrow \emptyset$ // Set of outlier clusters
for $i \leftarrow 1,\ldots,n$ **do**
    comp$\{i\} \leftarrow \{v_i\}$ // each node is in its own component
    $\boldsymbol{CC}[i] \leftarrow 1$ // each component has cardinality one
end for
$\boldsymbol{pred} \leftarrow 1,\ldots,n$ // each component is its own parent
for $k \leftarrow 1,\ldots,n-1$ **do**
    $(f_k, t_k) \leftarrow$ SpMST.pop() // Pop current shortest SpMST edge
    $c1 \leftarrow$ GETROOT(f_k, $\boldsymbol{pred}$) // component id of node f_k
    $c2 \leftarrow$ GETROOT(t_k, $\boldsymbol{pred}$) // component id of node t_k
    if $\boldsymbol{CC}[c1] + \boldsymbol{CC}[c2] < CC_{\max}$ **then**
        if $\boldsymbol{CC}[c1] < \boldsymbol{CC}[c2]$ **then**
            $\boldsymbol{pred}[c1] \leftarrow c2$ // Set $c2$ as parent of $c1$
            $\boldsymbol{CC}[c2] \leftarrow \boldsymbol{CC}[c2] + \boldsymbol{CC}[c1]$ // update cardinalities
            comp$\{c2\} \leftarrow$ comp$\{c2\} \cup$ comp$\{c1\}$
            comp$\{c1\} \leftarrow \emptyset$
            $\mathcal{C} \leftarrow$ comp$\{c2\}$ // the component to test
        else
            Repeats the operations in the *true* branch, but with $c1$ being the parent of $c2$.
        end if
        if $\phi(\mathcal{C}) < \phi_{\max}$ **then** $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{C}$ // score $\phi(\mathcal{C})$ is also stored
        end if
    end if
end for
Output: $\mathcal{S}$, outlier scores.

function GETROOT($node$, $\boldsymbol{pred}$) // Get component's id
    while $node \neq \boldsymbol{pred}[node]$ **do**
        $node \leftarrow \boldsymbol{pred}[node]$
    end while
return $node$
end function

As in the original Kruskal algorithm, the sorting order of the SpMST priority queue containing the SpMST edges is in the increasing order of the SpMST edge lengths. Thus, the union-find data structure, which is minimally represented by the component predecessor vector $\boldsymbol{pred}$, component size vector $\boldsymbol{CC}$, and the GETROOT function, is only used to keep track of the connected components resulting due to sequential addition of the SpMST edges between the nodes. The additional variable comp is used to keep

track of the current nodes in each connected component. It is shown in Algorithm 3.6 as a map from the component's index to the component's nodes, but this may be not the most efficient data structure. Finally, Algorithm 3.6 evaluates the outlier score of each newly obtained connected component and saves the component if the score is below the threshold.

Algorithm 3.7 Resolve cluster intersections

Input: Cluster outlier scores $\boldsymbol{\phi}$ (2.23), map to nodes of each cluster `clu`, cluster cardinalities $\boldsymbol{CC}$, initial number of clusters k.

1: Reorder the clusters `clu` and their sizes $\boldsymbol{CC}$ in the *increasing* order of the cluster outlier scores $\boldsymbol{\phi}$.
// Transform the node indices of each cluster into the cluster indicator row vectors combined into the matrix $\mathbf{T}_{k\times n}$.
2: **for** $i \leftarrow 1,\ldots,k, j \leftarrow 1,\ldots,n$ **do**
3: **if** $j \in$ `clu`$\{i\}$ **then** $t_{ij} = 1$
4: **end if**
5: **end for**
6: Set the status of each cluster indicator row of $\mathbf{T}$ to *true*.
7: $current \leftarrow 1$ // current row index
8: **while** status of some row of $\mathbf{T}$ is *true* **do**
9: Find clusters with indicator rows containing *any*, but not *all* ones in the non-zero columns of the *current* row. These clusters partially intersect with the *current* one.
10: Find clusters with indicator rows containing *all* ones in the non-zero columns of the *current* row and having the same entry in $\boldsymbol{CC}$ as the *current* cluster. These clusters are the duplicates of the *current* one.
11: Mark clusters discovered with the above two statements for the deletion. Set the corresponding rows of $\mathbf{T}$ to zero and the status of these rows to false.
12: Set the *current* row of $\mathbf{T}$ to zero and its status to false.
13: $current \leftarrow$ index of the topmost indicator row with the *true* status.
14: **end while**

Output: $\mathcal{S}$ as clusters in `clu` not marked for deletion, outlier scores (2.23) and (3.1).

AGGREGATION OF OUTLIER MINING RESULTS

It is common in large-scale networks to detect several outlier clusters that overlap with each other, and especially if the limits on outlier score (2.23) and outlier size were set rather large. While this situation occurs naturally (e.g., the same tightly clustered core group of nodes can be isolated with several cutsets of varying quality), it is often desirable to meaningfully resolve the cluster intersections. Thus Algorithm 3.7 is proposed, which aims to remove clusters that partially overlap with a cluster having a lower score (2.23). This is done because combining two non-fully overlapping clusters would induce a new cluster, the properties of which (e.g., the cardinality) are hard to control. However, Algorithm 3.7 would keep a less severe outlier cluster if it fully includes a more severe one, as the union of such clusters will still result in a valid cluster, with the smaller cluster being automatically integrated into the larger one. This integration step can be done by retrieving the edges running inside of each of the relevant clusters and using this set of

edges to find the subgraphs induced by these edges (e.g., with a connected components algorithm).

The two steps (Algorithm 3.5 and Algorithm 3.6) of the outlier mining process usually detect many identical clusters. Thus, Algorithm 3.7 is especially relevant in the context of combining the results of multiple outlier detection techniques.

OUTLIER MINING SUMMARY

After introducing Algorithms 3.4, 3.5, 3.6, 3.7, the proposed outlier mining process can be summarized as follows:

1. Select k as the dimension of spectral embedding and find the first k eigenvectors of the matrix $\mathbf{W_n}$.

2. Normalize the rows of the eigenvector matrix according to (2.14) and construct the embedded graph using the sets $\mathcal{V}$ and $\mathcal{E}$ of the original graph $\mathcal{G}$ and the spherical distances between the points in the spectral embedding [76].

3. Construct the SpMST as described in Algorithm 3.4.

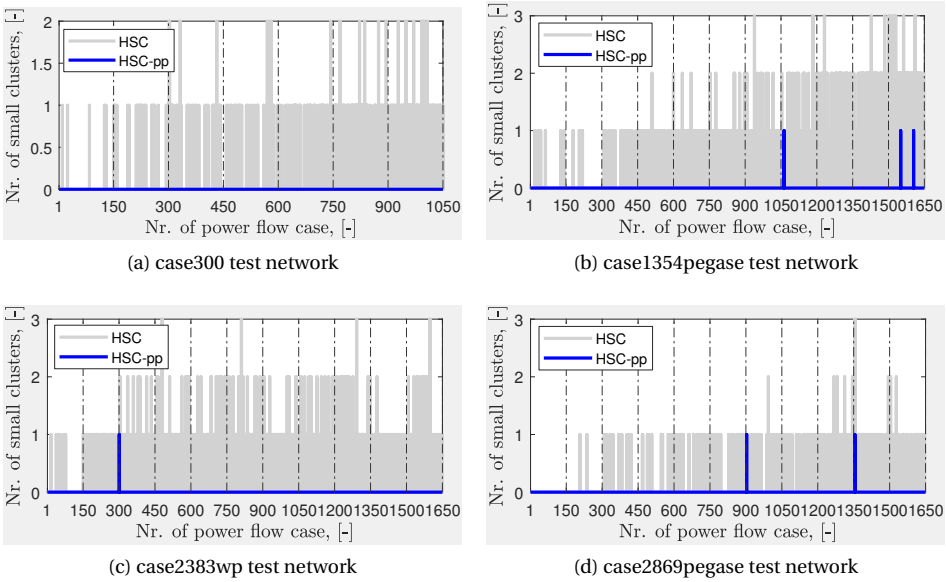
4. Use Algorithms 3.5 and 3.6 to detect the outlier cluster candidates.

5. Evaluate score (3.1) for every graph node to get the ranking of loosely connected single graph nodes. Mark a fraction of nodes with lowest outlier scores (3.1) as outliers.

6. Aggregate the identified outlier clusters and outlier nodes using Algorithm 3.7.

7. Process the obtained clusters and graph nodes as required by the application.

The first and last steps of the above list still require some clarification. Selecting a larger value of k is important if the method is needed to be more sensitive (i.e., to discover less severe outlier clusters). For detecting severest outliers, setting of k to 3 or 4 should be enough in the most cases. However, it will be shown in Section 3.4.3 that choosing higher values of k usually does not necessarily increase the processing time a lot. In what about the last step, the pre-processing of graph partitioning implies removing graph outliers in advance to prevent their separation by a spectral graph partitioning algorithm. This can be achieved by merging each outlier cluster into a single node as shown in Algorithm 3.3, but without updating the node weight of the aggregate node to the sum of node weights of the merged nodes. If the node weight is not updated, spectral clustering avoids to separate a single node, as this strongly increases the Ncut value (2.7). To further avoid the separation of single node outliers (including the newly formed merged nodes), they are merged with a neighbor node to which they have the strongest connection. The results in Section 3.4.3 demonstrate that this approach is quite efficient. In general, the processing of the returned outliers can vary depending on the application.



(a) case300 test network

(b) case1354pegase test network

(c) case2383wp test network

(d) case2869pegase test network

Figure 3.11: Occurrences of clusters below 10% of the average cluster size for the partitioning of randomly generated power flow graphs with an outlier-sensitive Algorithm 2.2, with (blue) and without (gray) pre-processing of potential small clusters.

3.4.3. OUTLIER MINING EVALUATION

To test the detection performance of the proposed graph outlier mining process in Section 3.4.2, it is applied as a pre-processing step for the partitioning of randomly generated active power flow graphs of the four networks from the MATPOWER toolbox [81, 162] ranging from 300 to 2869 nodes. The random power flow graphs are generated as explained in Section 3.2.2. The larger test networks *case1354pegase*, *case2383wp*, and *case2869pegase* are partitioned into $k = 2,\ldots,12$ clusters, while the smaller network *case300* is partitioned into $k = 2,\ldots,8$ clusters. Each number of clusters is tested on 150 unique randomly generated power flows. In addition, the computational performance of the proposed graph outlier mining is tested on the MATPOWER networks containing up to 13659 nodes.

The baseline clustering algorithm is HSC, which has been introduced in Section 2.5.2 as Algorithm 2.2. The used AHC linkage criterion is the average linkage. Using HSC was shown to produce high-quality clusters, while ensuring each cluster to be a connected subgraph [76]. However, hierarchical clustering is generally known to be sensitive to outliers if the number of clusters is given as an input [152], which is further confirmed by the test results in this section.

In this case study, the goal is to prevent the occurrences of clusters that are smaller than 10% of the average cluster size (i.e., with less than round$(0.1n/k)$ nodes) by applying outlier mining as graph pre-processing prior to graph partitioning. To achieve this goal, it is necessary to increase the outlier cluster threshold $\phi_{\max}$ as the number of clus-

ters is increased. Bi-partitioning the network aims to select the two most pronounced clusters. However, the clusters that are not distinct enough for the bi-partitioning may become important if a larger number of clusters is requested, which explains the need to increase $\phi_{\max}$ for the growing k. A heuristic expression $\phi_{\max} = 0.02 \log_2 k$ is used for the cluster outlier score threshold. For the *case2383wp* test network, reduction of too many clusters has caused the network to become over-constrained, which resulted in frequent small clusters with *high* values of score (2.23) (the values of 0.1–0.5 or larger reflect poor clusters). For this reason, the value of $\phi_{\max}$ for the *case2383wp* test network is constrained not to exceed 0.03. For the *case2869pegase* network, the value of $\phi_{\max}$ is set not to exceed 0.02 because it was acceptable not to raise it any higher for $k = 2, \ldots, 12$. For the single node outliers, the top 7% of candidates are kept instead of trying to guess a good absolute threshold for p_{max} (3.1). The number of eigenvectors to build the spectrally embedded graph for SpMST is chosen to be equal to the requested number of clusters k because of the requirement to detect quite subtle small clusters for the higher values of k.

As Figure 3.11 shows, the described pre-processing causes the HSC method to much less frequently return clusters below 10% of the average cluster size. While there still are a few occurrences of clusters below the specified size threshold, those could be avoided by adjusting the $\phi_{\max}$ value specifically for these test cases. Thus, it can be concluded that the proposed graph outlier detection method is able to reliably estimate the outlier clusters for a given graph (otherwise those clusters would be returned by the graph partitioning method). To provide more insight, the proposed graph outlier mining method is also used as a pre-processing for the NJW spectral clustering implemented as Algorithm 2.1. The aggregated results for the HSC and NJW methods are given in Table 3.12, with the † superscript denoting the versions of the methods with outlier pre-processing.

Test network	HSC	$HSC^{\dagger}$	NJW	$NJW^{\dagger}$
case300	253	0	13	0
case1354pegase	755	3	116	0
case2383wp	668	1	10	1
case2869pegase	231	2	1	0

Table 3.2: Total numbers of partitioning test cases with small cluster occurrences

To test the computational performance of the proposed graph outlier mining method, it is executed on the six MATPOWER networks ranging from 300 to 13659 nodes. For each network, the number of eigenvectors is varied from three to eight, while the upper cluster size limit is set to round$(0.1n/k)$ and the threshold of score (2.23) is set to 0.03. For this case study, the algorithm is run on the active power flow graphs obtained from the nominal loading profile of each network. All results were obtained on MATLAB R2017a (64-bit) on a PC with an Intel® Xeon® E5 3.70 GHz CPU and 16 Gb of RAM using a single core.

Surprisingly, choosing a low value for the number of eigenvectors k may result in longer running times. This can be explained by the complex convergence mechanisms of eigenvalue solvers for sparse matrices (e.g., of the Lanczos algorithm [70]). Nevertheless,

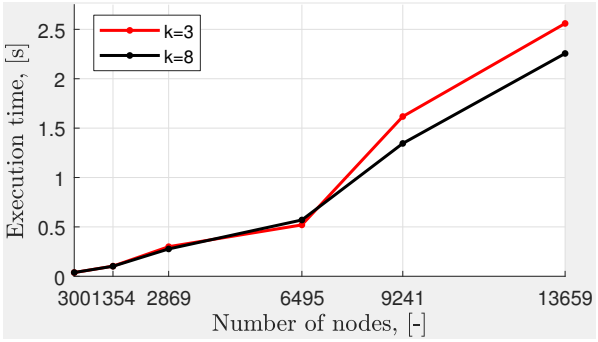


Figure 3.12: Execution time of the SpMST-based graph outlier mining method for the varying number of network nodes and dimension of spectral embedding

the execution times in Figure 3.12 justify the practicality of the proposed graph outlier detection method at least for the networks of several thousands of nodes.

The main computational requirement of the SpMST graph outlier detection method is in computing several largest eigenvalues and eigenvectors of the matrix $\mathbf{W_n}$. This procedure is reported to be tractable for sparse graphs of at least tens of thousands of nodes [152]. In practice, computing the eigenvectors took around 50–70% of the total execution time for each network using the highly optimized eigenvalue solver available in MATLAB. Such disproportion in the computation time can also be explained by the efficient algorithmic design of the SpMST-based outlier mining described in Section 3.4.2. The disparity between the eigenvector computation time and the SpMST processing time could be even larger if the latter component were also implemented in a high-performance compiled language instead of the MATLAB-based implementation.

3.5. CONCLUSIONS

This chapter considered some of the practical drawbacks of the mainstream graph partitioning methods and advocated the use of auxiliary pre- and post-processing algorithms as a possible solution. Subsequently, graph partitioning was represented as a three stage process including the pre-processing, partitioning and post-processing steps, which was illustrated in Figure 3.1.

The developments of this chapter were aiming at the three major objectives. First, the discussion about the inherent drawbacks of graph partitioning allowed to provide a deeper illustration of the graph partitioning problem that is central to this thesis. Second, discussing cluster connectivity and graph reductions and providing the solutions to these problems in the form of Algorithms 3.1 and 3.3 is important for the next chapters. Third, the case studies with Algorithms 3.2, 3.5 and 3.6 demonstrated the ways to improve the performance of graph partitioning methods focused on maximizing graph cluster separation. In particular, it has been shown that using the label propagation based graph cut improvement very often allowed to substantially improve the results of the NJW spectral graph partitioning without a significant effect on the cluster size balancing. The improvement could be less pronounced if a more sophisticated parti-

tioning algorithm was used instead of NJW, but the fast execution time and full independence from the chosen graph partitioning method justify the use of Algorithm 3.2 (or its enhanced variation) even in such cases. The use of graph outlier mining from Section 3.4 as partitioning pre-preprocessing was shown to substantially improve the clustering balance of outlier sensitive graph partitioning algorithms. The ability to detect loosely connected nodes and small clusters below certain size can potentially be helpful for applications other then balanced partitioning of power networks, although this possibility was not further developed inside the scope of this thesis.

To summarize, the use of auxiliary pre- and post-processing algorithms was shown to be an efficient approach to achieve an improved performance and higher flexibility of graph partitioning. The studies in this chapter also contributed to research questions I, II, and III that were stated in Section 1.2.3.

4

ORTHOGONAL STRUCTURE OF SPECTRAL EMBEDDING

4.1. INTRODUCTION

The main motivation of this chapter is the problem of choosing the proper number of zones or areas for power system partitioning, which has been stated as a research question in Section 1.2.3. In the power system literature, a significant number of papers have raised this problem as well. For example, Lagonotte et al. suggest to choose the number of VCZs by using the variation of slope of the relative diameter of VCZs versus the number zones, with hierarchical clustering being used for grouping [46]. In [52], the number of VCZs is chosen based on the maximal average inter-cluster distance, which is defined from hierarchical clustering. Another line of work [30, 31, 163] does not explicitly tackle the selection of the number of clusters, but acknowledges it as a relevant issue. This same issue is also relevant for many generator coherency algorithms [50, 121, 164]. Additionally, there are multiple power system applications outside of the scope of this thesis that depend on the proper selection of the number of clusters (e.g., contingency clustering [117]). Therefore, the above glimpse on the existing literature is only meant to introduce the topic, but not to fully cover it[1].

As it can be seen, the existing techniques to select the number of clusters often rely on the results of AHC (e.g., inter-cluster distances). However, the AHC results are not unique, as AHC utilizes various linkage criteria (e.g., single-link, complete-link etc.), and the resulting clusters as well as the corresponding intra- or inter-cluster distances can vary a lot depending on the chosen linkage criterion. Another popular approach to select the number of clusters is based on computing spectral eigengaps [34, 76, 94]. The early uses of this approach included the selection of slow electromechanical modes for generator slow coherency [34] (see (2.20)), and it gained more popularity as spectral clustering began to be more widely applied to power system problems. The spectral eigengap heuristic is based on the second property of the graph matrices used for spectral

[1] The material of this chapter is based on [J1]

clustering (see Section 2.5). That is, having k eigenvalues that are close to 0 (if $\mathbf{L}$, $\mathbf{L_{rw}}$, or $\mathbf{L_n}$ are used) or 1 (if $\mathbf{P}$ or $\mathbf{W_n}$ are used) indicates that the graph representing the clustered data is almost disconnected into k parts (i.e., k highly distinct clusters are present). Unfortunately, the occurrences of distinct large eigengaps (or relative eigengaps as in [76, 94]) are rare in many practical situations [129].

This chapter details an alternative approach to the selection of the number of clusters that is based on the alignment of the row-normalized spectral embedding (2.14) with the standard basis of the $\mathbb{R}^k$ Euclidean space. The advantage of this method lies in its independence from the internal mechanics of any clustering algorithm, which is a drawback of the popular heuristics based on AHC distances or the silhouette heuristic [71, 164, 165]. Instead, the analysis is performed on the row-normalized spectral embedding, which can be considered as inherent to a given weighted undirected graph. Unlike the spectral eigengap heuristic, the proposed method to evaluate the number of clusters directly relates to the expected clustering quality. To strengthen this point, a new graph partitioning algorithm is introduced that relies on the axes-aligned row-normalized spectral embedding to produce partitionings of high quality.

The rest of the chapter is organized as follows. Section 4.2 introduces the core ideas from [122, 129] which form the basis for the developed methods. Section 4.3 introduces the robust algorithm for spectral embedding alignment and justifies its correctness by comparing it with an alternative idea based on gradient descent (GD) minimization. Section 4.4 details the novel graph partitioning algorithm that further validates the value of axes-aligned spectral embeddings by showing their role in producing high quality graph partitionings. The high-quality of partitionings obtained with the new algorithm is demonstrated in Section 4.5 by performing comparisons with three other well-known methods: NJW, HSC, and Graclus [124]. Finally, Section 4.6 concludes the chapter.

4.2. ORTHOGONAL INVARIANCE OF SPECTRAL CLUSTERING

As it has been shown in Section 2.5, the optimal solution of the continuous spectral relaxation of the NP-complete normalized cut problem (2.8) is given by the lowest k eigenvectors of the matrix $\mathbf{L_{rw}}$ or the highest k eigenvectors of the matrix $\mathbf{P}$ (see Equations (2.10) and (2.11)). In their seminal work, Yu and Shi [122] emphasized that this optimal solution is invariant with respect to orthogonal linear transformations (i.e., rotations and reflections) applied to the initial eigenvector matrix $\mathbf{V}$ (2.10) or to its row-scaled version $\mathbf{U}$ (2.11). That is, the continuous optima of (2.11) form a subspace characterized as

$$\{\mathbf{UR} : \mathbf{R}^T\mathbf{R} = \mathbf{I}_k\} \tag{4.1}$$

where $\mathbf{U}_{n\times k}$ are the eigenvectors of $\mathbf{W_n}$ and $\mathbf{R} \in \mathbb{R}^{k\times k}$ is an arbitrary orthogonal matrix. In (4.1), the eigenvectors of $\mathbf{W_n}$ are used instead of the eigenvectors of $\mathbf{P}$ as in [122], as these eigenvectors can be computed more efficiently (see Section 2.5.1).

The invariance of continuous optima of (2.10) and (2.11) with respect to orthogonal linear transformations implies the possibility to select continuous optima from subspace (4.1) that facilitate the discretization of the eigenvector matrices $\mathbf{V}$ and $\mathbf{U}$ by revealing more information contained in them. To further facilitate the discretization of eigenvector matrices, the authors of [122] propose to normalize the rows of $\mathbf{U}$ to unit

length according to (2.14) and to seek the discrete solution $\widetilde{\mathbf{X}}$ by applying orthogonal linear transformations to the matrix $\mathbf{X}$. In the discrete indicator matrix $\widetilde{\mathbf{X}}$, each row represents a unit vector aligned with one of the canonical coordinate axes. Therefore, the row normalization step in [79, 122] brings the eigenvector matrices $\mathbf{U}$ or $\mathbf{V}$ closer to discrete solutions $\widetilde{\mathbf{X}}$ (the row normalization of both $\mathbf{U}$ and $\mathbf{V}$ results in the same matrix $\mathbf{X}$).

Furthermore, the authors of [122] propose an algorithm to find an orthogonal transformation $\mathbf{R}^*$ that would facilitate the discovery of a good discrete solution $\widetilde{\mathbf{X}}$ from the initial row-normalized optimal solution of the continuous relaxation $\mathbf{X}$. The discretization algorithm is stated as the optimization problem:

$$\begin{aligned} \text{minimize} \quad & Q(\widetilde{\mathbf{X}}, \mathbf{R}) = \|\widetilde{\mathbf{X}} - \mathbf{X}\mathbf{R}\|_F \\ \text{subject to} \quad & \widetilde{\mathbf{X}} \in \{0,1\}_{n\times k}, \widetilde{\mathbf{X}}\mathbf{1}_{k\times 1} = \mathbf{1}_{n\times 1}, \\ & \mathbf{R}^T\mathbf{R} = \mathbf{I}_k \end{aligned} \tag{4.2}$$

where $\|\cdot\|_F$ is the matrix Frobenius norm: $\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{k}|A_{ij}|^2}$. Objective (4.2) can be considered as a measure of closeness of the orthogonally transformed matrix $\mathbf{XR}$ to its discretized version $\widetilde{\mathbf{X}}$. Although the row-normalized matrix $\mathbf{X}$ is no longer a feasible solution to either (2.10) or (2.11) and not an eigenvector matrix in general, using it in (4.2) produces very good results in practice [123].

The formulation in (4.2) is in two unknowns: the discrete solution $\widetilde{\mathbf{X}}$ and the orthogonal matrix $\mathbf{R}$ that brings $\mathbf{X}$ closest to $\widetilde{\mathbf{X}}$. As there is no direct method to solve (4.2) simultaneously for both $\mathbf{X}$ and $\mathbf{R}$, an iterative procedure was proposed in [122].

If $\mathbf{R}$ is given in (4.2), $\widetilde{\mathbf{X}}$ is determined by *non-maximum suppression* on $\mathbf{Z} = \mathbf{XR}$, that is by setting the maximum entry of each row of $\mathbf{Z}$ to 1 and the remaining entries to zero, which is described by (4.3):

$$\widetilde{X}_{ij} = \begin{cases} 1, & \text{if } j = \operatorname{argmax}_{l\in\{1,\ldots,k\}} Z_{i,l} \\ 0, & \text{otherwise} \end{cases} \tag{4.3}$$

If $\widetilde{\mathbf{X}}$ is given in (4.2), $\mathbf{R}$ is determined by a singular value decomposition (SVD) of $\widetilde{\mathbf{X}}\mathbf{X}^T$:

$$\begin{aligned} \widetilde{\mathbf{X}}^T\mathbf{X} &= \mathbf{Q}\Sigma\mathbf{Y}^T \\ \mathbf{R} &= \mathbf{Q}\mathbf{Y}^T \end{aligned} \tag{4.4}$$

where $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_k)$ is the diagonal matrix of singular values of $\widetilde{\mathbf{X}}^T\mathbf{X}$ and $\mathbf{Q}$, $\mathbf{Y}$ are the matrices of the left and right singular vectors respectively.

The iterative approach in [122] to solve the problem in (4.2) consists of alternating the steps of optimal alignment (4.4) and non-maximum suppression (4.3) that rapidly converge to an initialization-dependent local optimum of the problem in (4.2). The proofs of optimality of (4.3) and (4.4) can be found in [122]. Noteworthy, the problem and the method in (4.4) to compute the orthogonal transformation $\mathbf{R}$ that most closely aligns $\mathbf{X}$ to $\widetilde{\mathbf{X}}$ is known as the *orthogonal Procrustes problem* [166].

By analyzing the cost function in (4.2), it is possible to notice that it has the goal of maximizing one entry per matrix row, while minimizing the remaining entries, by applying a single orthogonal transformation $\mathbf{R}$ on the input matrix $\mathbf{X}$. Geometrically this corresponds to the alignment of the initial spectral embedding with the axes of the canonical

coordinate system. A cost function that reflects the degree of alignment of a spectral embedding with the canonical coordinate system is further called *alignment cost.*

4.3. EIGENVECTOR ALIGNMENT AND NUMBER OF CLUSTERS

4.3.1. EIGENVECTOR BASED SELECTION OF NUMBER OF CLUSTERS

The concept of alignment cost was used by Zelnik-Manor [129] to select the number of eigenvectors that most closely resembles the ideal result of spectral clustering. The authors in [129] used the cost function in (4.5) that was formulated in terms of *unnormalized* eigenvectors $\mathbf{U}$:

$$\begin{aligned} &\text{minimize} \quad J(\mathbf{R}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}\frac{[\mathbf{UR}]_{ij}^2}{M_i^2} \\ &\text{subject to} \quad \mathbf{R}^T\mathbf{R} = \mathbf{I}_k \end{aligned} \tag{4.5}$$

where $M_i = \max_j [\mathbf{UR}]_{ij}$. The cost in (4.5) was minimized by optimizing the orthogonal matrix $\mathbf{R}$ with GD (see Appendix B). For this minimization, the orthogonal matrix $\mathbf{R}$ was restricted to be a *rotation* matrix. The best feasible minimum for the cost J is equal to one, and it is achieved when rotation $\mathbf{R}^*$ recovers a discrete matrix from $\mathbf{U}$. According to Section 2.5, this case corresponds to the best possible outcome of spectral clustering, as every node is perfectly assigned to one of the k clusters. This observation can be used to select the number of eigenvectors k that, after applying the orthogonal transformation $\mathbf{R}^*$, leaves the lowest ambiguity in the cluster assignment of the graph nodes.

Given the predefined feasible range of cluster numbers $k = k_{\min}, \ldots, k_{\max}$, the method [129] first computes the full set of eigenvectors $\mathbf{U}_{n\times k_{\max}}$. The optimization starts by computing the alignment for the first $k_{\min}$ eigenvectors and then proceeds by adding one by one the remaining columns from $\mathbf{U}$ to the previously aligned eigenvectors. The initialization from the previous alignment commonly speeds up the optimization and provides an initial upper bound on the alignment cost.

Alignment costs (4.5) can be minimized for the row-normalized matrix $\mathbf{U}$ as well (and in general for any matrix $\mathbf{M} \in \mathbb{R}^{n\times k}$). However, in this case it is not possible to simply add the next column to the previously aligned ones due to an additional condition that the Euclidean norm of each row should be equal to one. However, it is possible to use the previously computed transformations to initialize the next set of row-normalized eigenvector columns as follows:

$$\mathbf{R}^0_{k\times k} = \begin{bmatrix} \mathbf{R}^0_{k-1\times k-1}\mathbf{R}^*_{k-1\times k-1} & 0 \\ 0 & 1 \end{bmatrix} \tag{4.6}$$

where $\mathbf{R}^0_{k\times k}$ is the initializing orthogonal transformation for $\mathbf{X}_{k\times k}$ and $\mathbf{R}^*_{k-1\times k-1}$ is the best estimated orthogonal matrix for $\mathbf{X}_{k-1\times k-1}$. Thus, the initializing transformation (4.6) is computed by accumulating the solutions from the previously considered cluster numbers.

In what follows, the minimization of alignment costs (4.2) and (4.5) is going to be considered only for the row-normalized eigenvector matrices $\mathbf{X}$, as this choice has shown superior results in practice.

4.3.2. ROBUST ORTHOGONAL INITIALIZATION

By comparing the alignment costs in (4.2) and (4.5), it is possible to notice that they are both non-linear non-convex optimization problems aiming to align the rows of matrices $\mathbf{X}$ and $\mathbf{U}$ with the canonical coordinate system. Under the premise of an efficient optimization strategy, the global minima of (4.2) and (4.5) should be close to each other. However, the results of minimizing (4.2) and (4.5) by the algorithms proposed in [122, 129] may differ quite a lot in practice, as the local optima achieved by these methods (see also Sections 4.2, 4.3.1 and Appendix B) are dependent on the initial orientation of the spectral embedding.

To avoid the poor local optima, it is desirable to start the alignment cost minimization from a good initial point. The authors in [122] used the problem-specific initialization approach from [79], which is a fast greedy algorithm to find a set of k nearly orthogonal rows in a matrix. The clustering initialization algorithm from [79] is extended here to handle the two important issues:

1. Starting the initialization [79] only once may not lead to a good result. Situations are possible, when the set of nearly orthogonal initialization points lies close to a poor local optimum. Therefore, it is desirable to develop a systematic strategy for multiple initializations.
2. The rows found by the initialization [79] are generally not perfectly orthogonal to each other, thus the transformation matrix formed by those rows is not strictly orthogonal.

The above issues may have only a small impact when the data is well-separated (as in the test studies in [79]). In this case, the likelihood to find a set of k nearly orthogonal points close to the global optimum significantly increases. If the graph has many nodes (e.g., the large image graphs in [122]), the second issue also becomes less common. However, power networks do not generally possess the just discussed properties.

The proposed Algorithm 4.1 uses at most $i_{\max}$ restarts to robustify the initialization. The restart strategy selects the first vector of the next k-dimensional basis formed from the rows of $\mathbf{X}$ as the row of $\mathbf{X}$ that has the minimal cumulative *cosine similarity* to the first vectors of the previously selected bases. As the rows of $\mathbf{X}$ are normalized by (2.14), cosine similarity is equivalent to the dot product. Rows that are more similar to any previously selected first basis row than the threshold τ are constrained not to become the first row vector in a new k-dimensional basis.

The retrieved k rows of $\mathbf{X}$ (combined into the matrix $\mathbf{B}$ in Algorithm 4.1) may not form an orthonormal basis. For a set of linearly independent vectors, the closest orthonormal basis is given by the SVD-based Loewdin orthogonalization:

$$\begin{aligned} \mathbf{B} &= \mathbf{Q}\Sigma\mathbf{Y}^T \\ \mathbf{R} &= \mathbf{Q}\mathbf{Y}^T \end{aligned} \tag{4.7}$$

The Loewdin orthogonalization (4.7) is used to transform the retrieved k rows of $\mathbf{X}$ stored in the columns of $\mathbf{B}$ to a proper orthogonal transformation. Finally, Algorithm 4.1 evaluates the alignment cost associated with each obtained set of k rows of $\mathbf{X}$ and returns the best encountered aligning transformation. In addition, all discovered sets of k rows

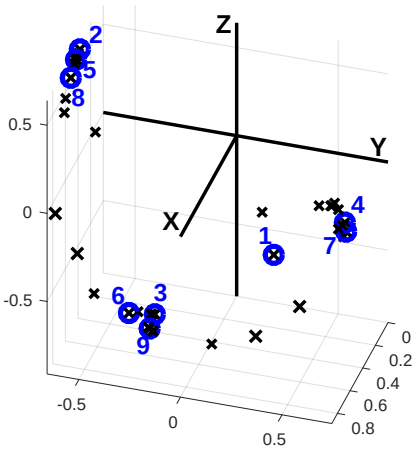


Figure 4.1: Selection of starting points by the initialization algorithm for spectral embedding alignment. The 3-spectral embedding is computed for the admittance graph in Figure 4.3. The circles represent start points, and the numbers represent their selection order.

Algorithm 4.1 Robust orthogonal initialization

Input: Row-normalized matrix $\mathbf{X}$, iteration limit $i_{\max}$, threshold τ

$\mathcal{S} \leftarrow \{1,\ldots,n\}$ // Rows of $\mathbf{X}$ eligible to start a basis
$\boldsymbol{s} \leftarrow \mathbf{0}_{n\times 1}$ // Cumulative cosine similarity
for $i \leftarrow 1,\ldots,i_{\max}$ **do**
  if $\mathcal{S} = \emptyset$ **then break end if**
  $r_1 \leftarrow \operatorname{argmin}_{l\in\mathcal{S}} \boldsymbol{s}[l]$ // Index of start row
  $\mathbf{B}[1,\ldots,k;1] \leftarrow \mathbf{X}[r_1;1\ldots,k]^T$
  $\boldsymbol{c} \leftarrow \mathbf{X}\mathbf{B}[1,\ldots,k;1]$
  $\mathcal{S} \leftarrow \mathcal{S} \setminus \{l \mid \boldsymbol{c}[l] > \tau\}$
  $\boldsymbol{s} = \boldsymbol{s} + \boldsymbol{c}$
  $\boldsymbol{c} = \operatorname{abs}(\boldsymbol{c})$ // Element-wise absolute value
  for $j = 2$ to k **do**
    $r_j \leftarrow \operatorname{argmin} \boldsymbol{c}$
    $\mathbf{B}[1,\ldots,k;j] \leftarrow \mathbf{X}[r_j;1\ldots,k]^T$
    $\boldsymbol{c} = \boldsymbol{c} + \operatorname{abs}(\mathbf{X}\mathbf{B}[1,\ldots,k;j])$
  end for
  $\mathbf{R} \leftarrow$ `loewdin`$(\mathbf{B})$ // (4.7)
  $Q \leftarrow$ Evaluate (4.2) with $\widetilde{\mathbf{X}}$ obtained from $\mathbf{X}\mathbf{R}$ with (4.3).
  Save Q and $\mathbf{B}$ obtained at each iteration.
end for
$Q^* \leftarrow$ The lowest Q
$\mathbf{R}^* \leftarrow \mathbf{R}$ that leads to Q^*

Output: Lowest alignment cost Q^*, best orthogonal transformation $\mathbf{R}^*$

are saved to subsequently provide multiple initializations for the algorithms to minimize the alignment costs (4.2) or (4.5).

A sample run of the restart strategy is illustrated in Figure 4.1. The first starting point is selected at random far from the three dense orthogonal clusters. However, the second starting point is selected in the top dense cluster, and all the following starting points are selected in the dense orthogonal clusters. In other words, the proposed restart strategy selects points representative for the orthogonal structure of spectral embedding. The similarity threshold parameter τ serves as a "step size" that prevents the subsequent starting points from being too close.

Although the proposed Algorithm 4.1 is based on the clustering *initialization* method [79], the added extensions make it a robust (i.e., not prone to poor local optima) method to minimize the alignment cost. The returned orthogonal transformation often needs only little improvement by the minimization methods from Sections 4.2, 4.3.1.

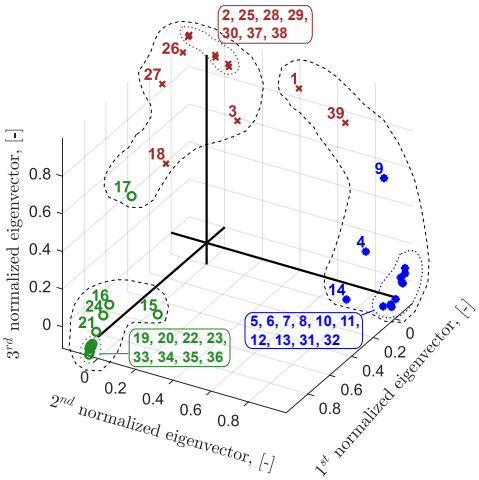


Figure 4.2: Aligned 3-spectral embedding for the partitioning study shown in Figure 4.3. The different colors and marker shapes represent the result of the NJW algorithm; the dashed lines show the grouping obtained by the partitioning algorithm from Section 4.4.

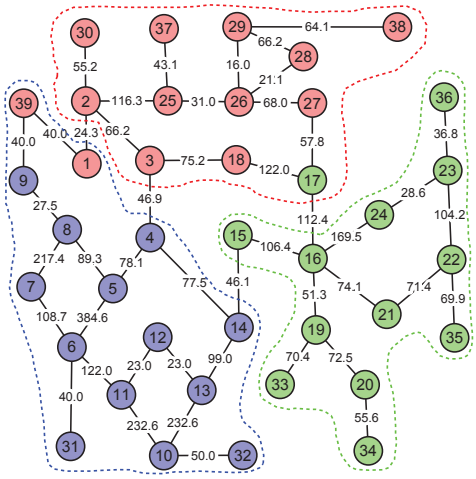


Figure 4.3: Branch admittances of the IEEE 39 bus test network, and spectral clustering into three parts with the NJW algorithm and the partitioning algorithm from Section 4.4. The areas found by the NJW algorithm are colored differently, and the boundaries of the areas found by the proposed partitioning method are shown with dashed lines.

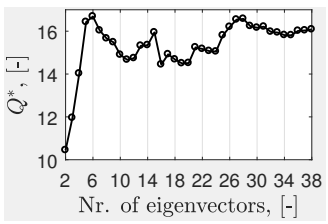
4.3.3. ALIGNMENT COST MINIMIZATION SUMMARY

Based on the information given in Sections 4.2, 4.3.1, and 4.3.2, a combined spectral embedding alignment algorithm can be formulated that consists of the following three steps:

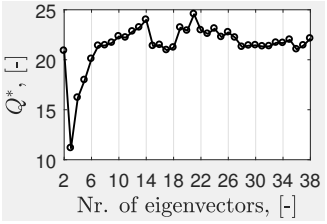
1. Initialization from previously aligned spectral embeddings.
2. Robust orthogonal initialization.
3. Final alignment cost minimization.

The overall philosophy of the proposed three-step algorithm is to apply several efficient methods to sequentially bound the alignment cost and reach a near-global optimum. At first, the alignment cost is reduced by applying the previous orthogonal transformations (accumulated in a matrix as illustrated by (4.6)) to the next set of row-normalized columns of $\mathbf{U}$. This step is mainly included due to its very low computational cost and ability to produce a quick initial bound of the optimization objective. The second step was described in Section 4.3.2. The third step is the iterative minimization of (4.2) or (4.5) from multiple starting positions. The multiple restart strategy of Section 4.3.2 supplies initializations of varying quality to the third step, resulting in the overall high-quality optimum.

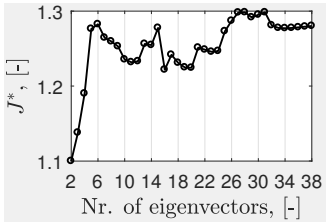
As a graphic illustration of the algorithm's possible outcome, Figure 4.2 shows the aligned 3-spectral embedding from which the partitioning result in Figure 4.3 has been obtained. Another illustration was already given above in Figure 4.4.



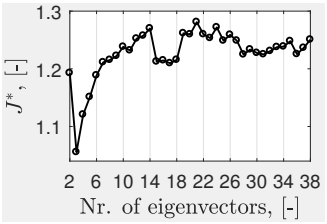
(a) case1354pegase test network



(b) case2869pegase test network



(c) case1354pegase test network



(d) case2869pegase test network

Figure 4.4: Minimized alignment costs (4.2) and (4.5) for branch admittance graphs of the two networks from MATPOWER with all transformer phase shifts set to zero.

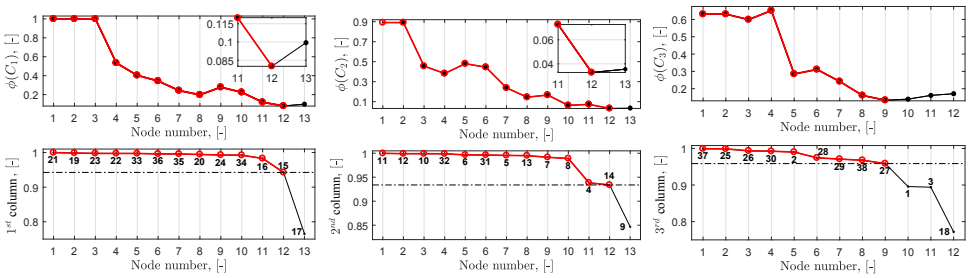
4.3.4. SELECTION OF ALIGNMENT COST

The spectral embedding alignment procedure of Section 4.3.3 can be used to robustly obtain near global minima of both (4.2) and (4.5). To illustrate this, the results of applying this procedure to the branch admittance graphs of the two large scale MATPOWER networks (*case1354pegase* and *case2869pegase*) are shown in Figure 4.4. As it can be seen, the minimization of alignment costs (4.2) and (4.5) discovers a very similar pattern.

The GD-based minimization of (4.5) uses Givens angles as optimization variables, which is explained in more detail in Appendix B. The number of Givens angles for k eigenvectors is equal to $k(k-1)/2$; i.e. the solution space grows quadratically with the increase of k. Therefore, the computation time of the gradient-based eigenvector alignment (4.5) showed to be noticeably higher, especially as the number of eigenvectors increased. Another issue with the gradient-based minimization is the necessity to properly choose the learning rate parameter. Thus, the minimization of objective (4.2) will be used in the following to discover the orthogonal structure of spectral k-embeddings.

The cost functions in (4.2) and (4.5) are largely motivated by the corresponding algorithms for their minimization. While (4.2) or (4.5) can be conveniently used to estimate the proximity of spectral embedding to the canonical coordinate axes, their numerical values cannot be easily related when comparing the alignment results for different input graphs (e.g., see the numerical values in Figure 4.4). To circumvent this issue, a new alignment cost metric is proposed:

$$C = \frac{1}{n} \operatorname{trace}\left(\widetilde{\mathbf{X}}^T \mathbf{X}^*\right) \tag{4.8}$$



(a) 1^{st} column of $\mathbf{X}^*$ (b) 2^{nd} column of $\mathbf{X}^*$ (c) 3^{rd} column of $\mathbf{X}^*$

Figure 4.5: Cluster core estimation from the axes-aligned spectral embedding in Figure 4.2 using Algorithm 4.2. Each column of $\mathbf{X}^*$ was sorted in descending order, and the starting size of cluster core was set at 2.

where $\mathbf{X}^* = \mathbf{X}\mathbf{R}^*$ is the final aligned spectral embedding obtained through the best aligning orthogonal transformation $\mathbf{R}^*$ and $\widetilde{\mathbf{X}}$ is obtained from $\mathbf{X}^*$ through non-maximum suppression (4.3).

The metric (4.8) is introduced solely for the evaluation and comparison purposes, while the actual spectral embedding alignment is obtained through the same methods that were discussed in the previous sections. Noteworthy, the metric (4.8) attains high values if the alignment of $\mathbf{X}$ is good, with the highest value being equal to 1. Therefore, the metric (4.8) has the meaning of *alignment quality* as opposed to alignment cost.

4.4. EIGENVECTOR ALIGNMENT AND GRAPH PARTITIONING

Apart from providing good indicators to select the number of clusters, the axes-aligned spectral embedding can also be a valuable input to partition the network. By looking at Figure 4.2, it is possible to see that some buses reside in dense *cluster cores*, while others (e.g. 1, 17, 18, 39) have their cluster membership less certain. If the computed aligning orthogonal transformation is denoted as $\mathbf{R}^*$ and $\mathbf{X} \in \mathbb{R}^{n\times k}$ is a row-normalized matrix of eigenvectors of $\mathbf{W_n}$, the axes-aligned normalized spectral embedding $\mathbf{X}\mathbf{R}^*$ can be referred to as $\mathbf{X}^*$. With aligned $\mathbf{X}^*$, a *cluster core* becomes numerically recognizable as the corresponding entries of some column of $\mathbf{X}^*$ will be close to one. And because the Euclidean norm of each row equals to one, the entries of the remaining columns in the same row will be close to zero.

The cluster core estimation process is formulated as Algorithm 4.2. First, the columns of $\mathbf{X}^*$ are sorted individually to reveal which of their rows have a large magnitude. Then the cluster core is initiated with the original row indices of the first n_{min} entries of the sorted column of $\mathbf{X}^*$, where n_{min} is obtained from the cluster size requirement (2.25). The next nodes are added to the core in the decreasing order of the corresponding column entries until the predefined threshold β is reached. To consider each column of $\mathbf{X}^*$ independently, this threshold value should be above $\sqrt{2}/2$. The value $\sqrt{2}/2$ ensures that no two (or more) columns of $\mathbf{X}^*$ can simultaneously assign the same row to their clusters. The initial value of β has been set to $\sqrt{3}/2$ for all the results in this chap-

Algorithm 4.2 Cluster cores from axes-aligned spectral embedding

Input: Aligned normalized spectral embedding $\mathbf{X}^*_{n\times k}$, matrix $\mathbf{W}$, threshold β, minimal number of nodes per cluster n_{min}

$\mathcal{CC} \leftarrow \emptyset$ // cluster cores
for $j = 1$ to k **do**
  $\boldsymbol{X} \leftarrow \mathbf{X}^*[1,\ldots,n;j]$ // j^{th} column of $\mathbf{X}^*$
  $\boldsymbol{ord} \leftarrow$ Descending order of entries in $\boldsymbol{X}$
  if $\max(\boldsymbol{X}) - 0.1 < \beta$ **then**
    $\beta_x \leftarrow \max(\max(\boldsymbol{X}) - 0.1, \sqrt{2}/2)$ // If current $\boldsymbol{X}$ is not well-aligned, use lower β
  else
    $\beta_x \leftarrow \beta$
  end if
  $\boldsymbol{core} \leftarrow \boldsymbol{ord}[1,\ldots,n_{min}]$
  $\boldsymbol{phi}[1,\ldots,n_{min}] \leftarrow \phi(\mathbf{W},\boldsymbol{core})$ // (2.23), (4.9)
  $i \leftarrow n_{min}+1$
  while $\boldsymbol{X}[\boldsymbol{ord}[i]] \geq \beta_x$ **do**
    $\boldsymbol{core} \leftarrow \boldsymbol{ord}[1,\ldots,i]$
    $\boldsymbol{phi}[i] \leftarrow \phi(\mathbf{W},\boldsymbol{core})$ // (2.23)
    $i \leftarrow i+1$
  end while
  $i^* \leftarrow \operatorname{argmin} \boldsymbol{phi}$
  $\mathcal{CC} \leftarrow \mathcal{CC} \cup \{\boldsymbol{core}[1,\ldots,i^*]\}$
end for

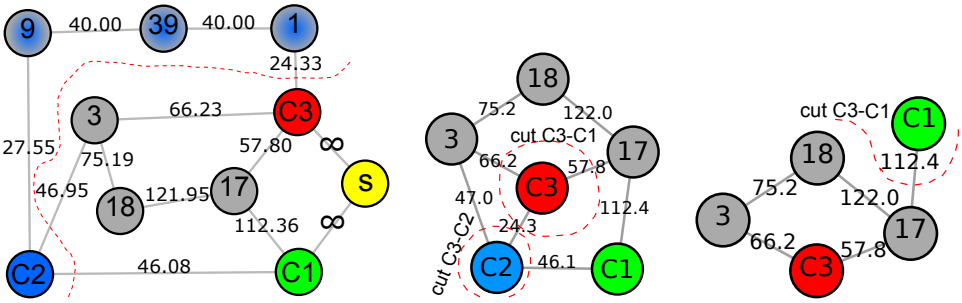
Output: $\mathcal{CC}$ // The k cluster cores

ter, which guarantees the other column entries for the same node not to exceed 0.5. The expansion ratio of the cluster core is updated after adding each next node, and the final cluster core is selected as the set of nodes with the smallest achieved expansion. In the majority of cases, the minimal expansion ratio corresponds to a cluster core having a single connected component. If there are multiple connected components, the next smallest expansion with the index higher than n_{min} is accepted, and the connectivity is checked for the corresponding group of nodes. In the worst case, the largest connected component can be taken as the core. However, such situations are not common in practice and mostly occur when the alignment of normalized spectral embedding is poor. An example of the cluster core estimation approach is shown in Figure 4.5 for the partitioning of the admittance graph of the IEEE 39 test network (see Figure 4.3).

The expansion ratio in Algorithm 4.2 can be updated efficiently by using an alternative formula:

$$\phi(\mathcal{C}) = \frac{\mathrm{vol}(\mathcal{C}) - \mathrm{links}(\mathcal{C},\mathcal{C})}{\mathrm{vol}(\mathcal{C})} \tag{4.9}$$

In (4.9), the cluster volume $\mathcal{C}$ can be updated by adding the pre-computed degree of the new cluster node (2.1) to the cluster volume value at the previous iteration. The weight of internal cluster edges $\mathrm{links}(\mathcal{C},\mathcal{C})$ can be updated by adding the double weight



(a) Improvement of cluster core C2 (b) Final bisection (step 1) (c) Final bisection (step 2)

Figure 4.6: Use of minimum s-t cuts for cluster core improvement and final recursive bisection

of the edges linking the new cluster node with the cluster nodes at the previous iteration (i.e., $2\sum_{i\in\mathcal{C}} W_{ij}$). Implementing these steps makes it possible to compute the many expansion ratio values in Algorithm 4.2 with a minimal computational overhead.

After all cluster cores have been estimated, they are improved one-by-one in the decreasing order of their expansion ratios:

1. Rerun the cluster core estimation Algorithm 4.2 with the threshold β only slightly above of $\sqrt{2}/2$.
2. Merge each cluster core to a single *core node* and find the minimum isolating s-t cut from the current core node to the remaining core nodes. A new fictitious sink node should be created and connected to the remaining core nodes with edges of an infinitely large weight. Then the isolating cut is computed as the minimum s-t cut between the current core node and the fictitious sink node [109]. Increase the current cluster core by the nodes that reside on its side of the cut.

The final improvement is chosen as one that reduces the expansion ratio most. The goal of this procedure is to decrease the objective (2.24) by greedily attempting to reduce the expansion ratios of the least fit cluster cores. The use of minimum s-t cuts (i.e., solutions to the max-flow/min-cut problem [108]) is motivated by their ability to rapidly find the globally optimal cut between two nodes (or two sets of nodes) in the graph. The classical drawback of minimum graph cuts to return highly unequally-sized bisections [74] is circumvented here by looking for the minimum cut that separates a whole cluster core (merged into a node) from the remaining core nodes. An example of use of minimum s-t cuts for cluster core improvement is shown in Figure 4.6a.

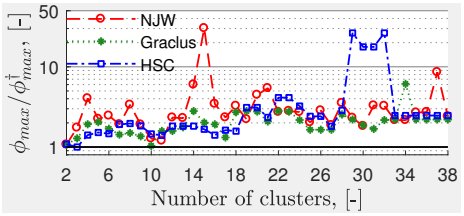
After the refinement stage, the updated cluster cores are once again collapsed into single nodes. The reduced network should consist of k core nodes and all the remaining nodes that were not assigned to the cluster cores, and it is partitioned via *recursive bisection*. At every step of recursive bisection, candidate minimum s-t cuts [108] are computed between an arbitrary core node and the remaining ones, and the lowest of these cuts is retained. This process iterates until all core nodes become separated from each other with all the remaining nodes being assigned to a cluster. The resulting partitioning is guaranteed to be connected, as cluster cores were constrained to be connected, and

minimal s-t cuts always separate the input graph into two connected parts. For example, Figure 4.6b shows the two candidate minimum s-t cuts with cluster core C3 being the source and cluster cores C1 and C2 being the two targets. As the value of cut C3-C2 is lower, it is retained, and the final partitioning is obtained by computing the s-t cut C3-C1 in the residual network resulting after the removal of node C2, as shown in Figure 4.6c.

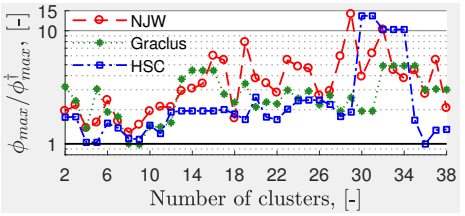
4.5. EVALUATION OF GRAPH PARTITIONING ALGORITHM

To evaluate the proposed graph partitioning method, it is tested on the branch admittance graphs of the two networks from the MATPOWER toolbox [81, 140] for which the alignment cost plots are presented in Figure 4.4. No modifications (e.g., reduction of leaf nodes [30, 94]) have been performed on the networks, except fixing the control angles of the few available phase shifting transformers at zero degrees to preserve the symmetry of the graph adjacency matrix. However, a phase shifting transformer with a non-zero phase shift can be represented in DC power flow by an equivalent (symmetric) admittance (see [167]), thus potentially allowing an extension of spectral graph partitioning to power networks with phase shifting transformers.

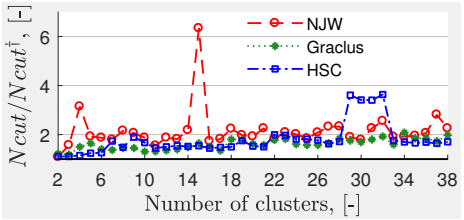
The partitioning algorithm in Section 4.4 is compared with the NJW and HSC algorithms from Section 2.5.2, and with the multilevel kernel k-means software Graclus [124]. The chosen hierarchical clustering linkage criterion of the HSC method is average linkage, as it was producing consistently better results. As Graclus and the NJW algorithm do not generally guarantee connected partitions, Algorithm 3.1 is used to ensure the cluster connectedness in these two cases. To provide a fair comparison, Algorithm 3.2 is not used to additionally refine the results of any of the four tested methods.



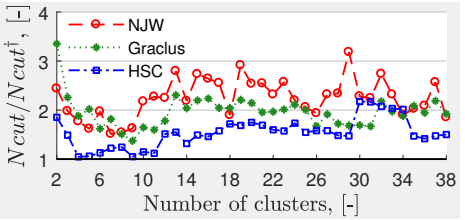
(a) case1354pegase test network



(b) case2869pegase test network

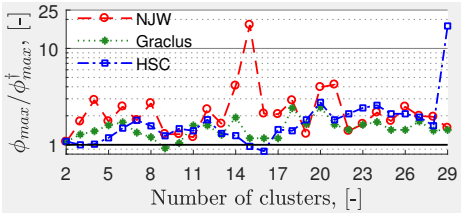


(c) case1354pegase test network

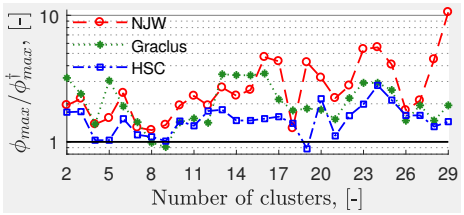


(d) case2869pegase test network

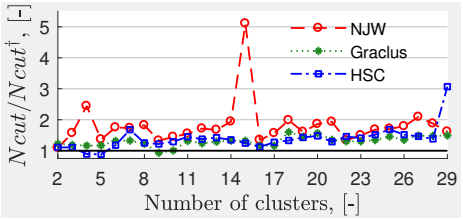
Figure 4.7: Comparison of maximal expansion ratio and normalized cut for partitioning of the branch admittance graphs of the two test networks from the MATPOWER toolbox



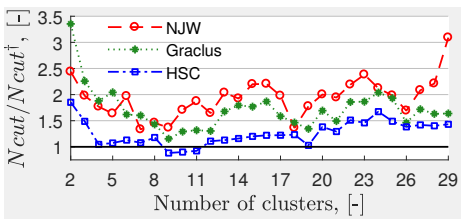
(a) case1354pegase test network



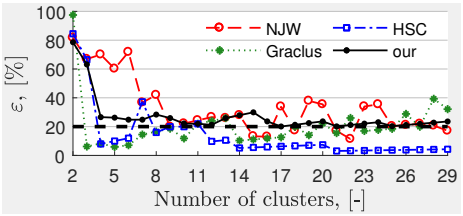
(b) case2869pegase test network



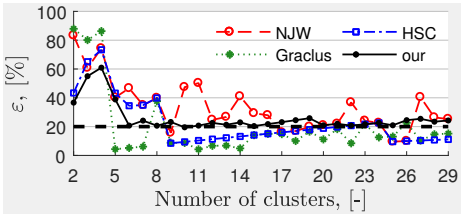
(c) case1354pegase test network



(d) case2869pegase test network



(e) case1354pegase test network



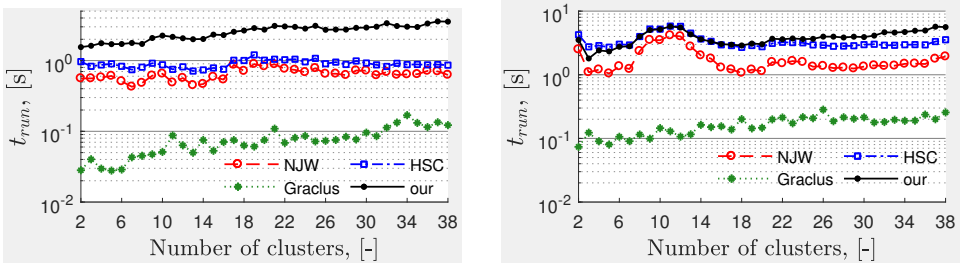
(f) case2869pegase test network

Figure 4.8: Partitioning of the branch admittance graphs of the two test networks from the MATPOWER toolbox with the minimal cluster size constraint of 20% of the average cluster size

The maximal expansion ratio (2.24) as well as normalized cut (2.7) are normally increasing in magnitude with the growing number of clusters. To compare the partitioning performance for different numbers of clusters on the same scale, it is decided to show the ratios of the results of the three test methods to the result of the algorithm in Section 4.4 (denoted by the † superscript). In addition, the logarithmic y-axis is often used to make the data on the plots more separable.

For the purpose of comparison, the minimal cluster size is initially set to $\lceil 0.03n/k \rceil$, with $k = 2,\ldots,38$ being the requested number of clusters. This small value is chosen to enable a fair comparison between the algorithms, as neither of NJW, Graclus, and HSC allows the specification of the minimal cluster size. The maximal number of clusters $k_{\max}$ is set to a relatively high value of 38 to demonstrate that the proposed partitioning method generally shows a good performance both for few and many clusters.

For the case of unconstrained cluster sizes, Figure 4.7 demonstrates that the partitioning algorithm based on the orthogonal structure of spectral embedding outperforms the HSC method [76] in the majority of the cases, and often by a large margin. At the



(a) case1354pegase test network

(b) case2869pegase test network

Figure 4.9: Total partitioning time of the four tested algorithms

same time, the HSC method with average linkage can be considered as an efficient partitioning algorithm, as it usually performs noticeably better than Graclus and NJW.

Figure 4.8 demonstrates the test results for the case when all clusters are required to be not smaller than 20% of the average cluster size n/k. The information provided by the aligned spectral embedding about the approximate locations and sizes of both small and large clusters in the network allows to neglect the columns of $\mathbf{X}^*$ describing presumably small clusters. As the result, the cluster size constraint is satisfied in all cases (see Figures 4.8e–4.8f). Satisfying the cluster size constraint has the associated cost in terms of partitioning quality: the HSC method now shows better ϕ_{max} and $Ncut$ more often, as it only aims to find compact clusters without imposing additional constraints on their size, as seen from Figures 4.8e–4.8f.

The computational time of the four algorithms for the largest tested number of clusters is shown in Figure 4.9. These results were obtained on MATLAB R2017a (64-bit) on a PC with an Intel® Xeon® E5 3.70 GHz CPU (single core computation) using a Linux virtual machine with 2 Gb of RAM. The use of the virtual machine running under Linix was due to the need to run Graclus. The computational time of the partitioning method from Section 4.4 includes the eigenvector computation time, spectral embedding alignment time, time to estimate and refine cluster cores and time of final recursive bisection. As it can be seen, the run time of the proposed partitioning method is slower than the HSC run time for the smaller 1354 bus test network, but this relationship improves as the network size increases. The non-monotonic increase of the computational time of the three eigenvector-based clustering methods (see Figure 4.9b) can be attributed to the complex convergence mechanisms of eigenvalue solvers for sparse matrices, as it has already been mentioned in Section 3.4.3.

4.6. CONCLUSIONS

This chapter has proposed an efficient methodology to select a good number of clusters when partitioning electric power networks, which constitutes research question V (see Section 1.2.3). Unlike several previous ideas [46, 52, 164, 165], the proposed method, which combines and extends the ideas from [122] and [129], does not depend on the internal mechanics of any particular clustering algorithm. Instead, it aims to explore

the structure of the graph spectral embedding, which can be considered as an inherent property of the studied graph. To this end, an optimal orthogonal transformation is estimated in a robust way to align the k-dimensional spectral embedding with the canonical coordinate system. The proposed method for the estimation of the number of clusters has been tested with the two alignment cost functions from [122] and [129] to reveal the nearly identical shape of the both objectives for the varying number of clusters. Additionally, a new quality function has been proposed to measure the average deviation of the spectral embedding from the canonical coordinate system.

To further justify the effectiveness of the proposed spectral embedding alignment scheme, an efficient k-way spectral partitioning algorithm has been proposed that uses the axes-aligned spectral embedding to find a set of k well-separated network clusters. This algorithm often achieves good results even if the spectral embedding does not show a distinct orthogonal structure, which may be explained by the great value of the global information about the locations of good clusters in the network that is contained in the axes-aligned spectral embedding. The possibility to approximately estimate cluster sizes allows to introduce a constraint on the minimal number of nodes per cluster, which represents a convenient way to avoid small clusters. Thus, the proposed k-way partitioning algorithm contributes to research questions I, II, while being able to ensure cluster connectedness (research question III). The described findings have been confirmed by the test results on partitioning of branch admittance graphs of the two large-scale power networks.

The results of this chapter mainly focus on the formulation of the new methods and their comparative testing using branch admittance graphs that are readily available from the network power flow data. As the next chapter will demonstrate, the methodology formulated in this chapter can potentially be used for a broad range of power system applications. For the practical applications of the proposed clustering methodology, it is important to remember about the clustering granularity issue [168]: for many datasets multiple "good" clusterings are possible depending on the cluster sizes that are looked for (i.e., the clustering resolution). Therefore, the highest spectral embedding alignment quality may be not the only criterion to select the final clustering structure. Instead, it may be better to select a different local maximum on the alignment quality plot if the corresponding clustering better satisfies the criteria imposed by the application at hand.

5

PARTITIONING FOR SVC AND GENERATOR SLOW COHERENCY

5.1. INTRODUCTION

This chapter focuses on the application of the methodology formulated in Chapter 4 to the two relevant power system problems: network zone division for SVC and identification of coherent groups of generators. The fundamentals of both of these applications have been discussed in Sections 2.2.1 and 2.6 respectively[1].

Regarding SVC, a zoning model is introduced that reformulates the earlier proposed VCS method [52] as a graph partitioning problem. The reformulation allows to apply the methods introduced in Chapter 4 to resolve some of the fundamental issues associated with SVC (most notably, the selection of the optimal number of pilot buses). This is achieved through evaluating the possible SVC zone divisions that show a high spectral embedding alignment quality and selecting the ones that result in a robust voltage regulation while using a moderate amount of pilot buses. Subsequently, the proposed SVC zoning methodology is verified by the two case studies involving the IEEE 39 bus test system and the IEEE 68 bus test system.

Similarly to SVC, the proposed generator slow coherency methodology is based on the spectral clustering framework introduced in Chapter 4. It is shown that when the eigenvectors of the electromechanical state matrix in (2.19) are *mass-normalized* (see [169, 170]), the orthogonal structure described in Chapter 4 emerges. The topic is concluded by the case studies on the NPCC 48 machine system.

This chapter is subdivided into two parts. First, Section 5.2 introduces and validates the new SVC zoning model. Second, Section 5.3 explores the links between generator slow coherency and the spectral clustering framework of Chapter 4 to propose and verify a new approach to generator coherency grouping. Section 5.4 aims to summarize the chapter, draw conclusions and provide an outlook.

[1] The material of this chapter is based on [J1] and [J3].

5.2. NETWORK ZONE DIVISION FOR SVC

As it has been detailed in Section 2.2.1, defining the SVC structure through electric power system division into VCZs has a significant practical merit, which justifies the use of this approach in multiple practical power systems [46, 47, 52, 93, 171].

Typically, power flow voltage sensitivities are involved into the VCZ definition, as these sensitivities precisely reflect the voltage relationships in the network. For example, inverted power flow sensitivity matrices of the form $\left[\frac{\partial Q}{\partial V}\right]^{-1}$ have been used to define the SVC structure of the French and Italian power systems [46, 47].

The VCS method, which has been used for SVC in China [52], uses a different voltage sensitivity $S'_{ij} = \frac{\partial V_{L,i}}{\partial Q_{G,j}}$, where $V_{L,i}$ is the voltage of the i load bus and $Q_{G,j}$ is the reactive power output of the j control generator. In the original VCS method, the sensitivities S'_{ij} are used to calculate the coordinates x_{ij} of the i load bus in the g-dimensional Euclidean space:

$$x_{ij} = -\log(|S'_{ij}|), \quad i = 1,\ldots,n,\ j = 1,\ldots,g \tag{5.1}$$

5

where g represents the number of control generators. After each bus receives its g-dimensional coordinate, the buses are clustered using AHC with the average inter-cluster distance serving as guidance to select the number of VCZs [52, 94]. In what follows, an alternative formulation of the VCS method is detailed, with the above description serving as an introduction and reference.

5.2.1. ZONING MODEL

The voltage sensitivities that are used for SVC zoning can be derived from the power flow Jacobian matrix containing the partial derivatives of active and reactive powers w.r.t. the voltage magnitudes and angles. The required expression can be written as:

$$\boldsymbol{\Delta Q} = \mathbf{B}\boldsymbol{\Delta V} \tag{5.2}$$

where $\boldsymbol{\Delta V}$ is the vector of small bus voltage magnitude changes, $\boldsymbol{\Delta Q}$ is the vector of small changes in bus reactive power injections, and $\mathbf{B}$ is the sensitivity matrix relating the two quantities.

In HV transmission power networks, reactive power injections have the major influence on bus voltages. Therefore, the matrix $\mathbf{B}$ is often modeled by the fast decoupled load flow (FLDF) matrix $\mathbf{B}''$ [172], which is equal to the negative nodal susceptance matrix. The accuracy of this simplified approach diminishes as the network becomes more heavily loaded. A more precise estimate of the matrix $\mathbf{B}$ is obtained by computing it from the power flow Jacobian corresponding to the actual operating condition by neglecting the effect of active power changes on voltage magnitudes. Setting the active power changes to zero allows to reduce the power flow Jacobian to a $n \times n$ matrix relating bus voltage magnitudes and reactive power injections. This matrix will be used for the sensitivity calculations in this chapter, and its derivation is illustrated in [1, Chapter 14], [95] and many other references. To compute voltage sensitivities between generator and load buses, the matrix $\mathbf{B}$ is expressed in the partitioned form:

$$\begin{bmatrix} \Delta \boldsymbol{Q_G} \\ \Delta \boldsymbol{Q_L} \end{bmatrix} = \begin{bmatrix} \mathbf{B_{GG}} & \mathbf{B_{GL}} \\ \mathbf{B_{LG}} & \mathbf{B_{LL}} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{V_G} \\ \Delta \boldsymbol{V_L} \end{bmatrix} \tag{5.3}$$

where $\Delta \boldsymbol{V_G}$ and $\Delta \boldsymbol{V_L}$ are vectors of voltage magnitude changes at generator and load buses respectively, $\Delta \boldsymbol{Q_G}$ and $\Delta \boldsymbol{Q_L}$ are vectors of changes of reactive power injections at generator and load buses respectively, and $\mathbf{B_{GG}}$, $\mathbf{B_{GL}}$, $\mathbf{B_{LG}}$, $\mathbf{B_{LL}}$ are the corresponding submatrices of $\mathbf{B}$. In the partitioned equation (5.3), the generator buses are assumed to coincide with PV and slack buses (i.e., the voltage-controlled buses), and the load buses are assumed to coincide with PQ buses.

To accommodate the VCS method to the spectral clustering methodology of Chapter 4, the relationships between control generators and load buses described by the sensitivities S'_{ij} are presented in the form of the *bipartite graph* with the symmetric adjacency matrix $\mathbf{S_Q}$ as follows:

$$\mathbf{S_Q} = \left[\begin{array}{c|c} \mathbf{0} & [S'_{ij}] \\ \hline [S'_{ij}]^T & \mathbf{0} \end{array}\right] \tag{5.4}$$

According to the references utilising the VCS method [173, 174, 175], the VCS sensitivities S'_{ij} in the matrix $\mathbf{S_Q}$ can be computed through the following steps:

1. Check the reactive power injections at generator buses. If the reactive power output of a generator is close to its upper limit, set its bus as a PQ node.

2. Set the j control generator as a PQ node. That is, the corresponding bus status in (5.3) changes from generator bus to load bus for a single iteration.

3. Invert the augmented matrix $\mathbf{B_{LL}}$ corresponding to the original PQ buses and the extra PQ bus of the j control generator.

4. From $\mathbf{B_{LL}^{-1}}$, extract the column vector corresponding to the bus of the j control generator, and in this vector keep only the entries corresponding to the original PQ buses. Insert this vector as the j column of the matrix $[S'_{ij}]$.

5. Repeat steps 1–4 for the remaining control generators.

In the above process, it is possible to substitute step 3 by slightly increasing the reactive power output of the j control generator and re-running the AC power flow. Then the sensitivities of load voltages to reactive power injections of the j control generator can be approximated through quotients of load bus voltage deviations by the reactive power increment of the j control generator.

The sensitivity of load voltages to generator voltage control can alternatively be stated as $S_{ij} = \frac{\partial V_{L,i}}{\partial V_{G,j}}$, where $V_{G,j}$ is the terminal voltage of the j control generator. It can expressed from the second set of equations in (5.3):

$$\Delta \boldsymbol{V_L} = \mathbf{B_{LL}^{-1}} \Delta \boldsymbol{Q_L} - \mathbf{B_{LL}^{-1}} \mathbf{B_{LG}} \Delta \boldsymbol{V_G} \tag{5.5}$$

From (5.5), the SVC sensitivity between generator and load voltages is given by:

$$\mathbf{S} = -\mathbf{B}_{\mathbf{LL}}^{-1}\mathbf{B}_{\mathbf{LG}} \tag{5.6}$$

and the corresponding bipartite graph is described by the symmetric adjacency matrix $\mathbf{S_V}$ as follows:

$$\mathbf{S_V} = \left[\begin{array}{c|c} \mathbf{0} & [S_{ij}] \\ \hline [S_{ij}]^T & \mathbf{0} \end{array}\right] \tag{5.7}$$

The matrices $\left[S'_{ij}\right]$ and $[S_{ij}]$ are related. Namely, the columns of $\left[S'_{ij}\right]$ are the scaled versions of the columns of $[S_{ij}]$ and vice versa. Using the matrix $\mathbf{S_V}$ instead of the matrix $\mathbf{S_Q}$ has the advantage of more regular graph weights: the sensitivities $\frac{\partial V_{L,i}}{\partial V_{G,j}}$ are effectively *voltage attenuations* [46] belonging to the range $[0,1]$. The sensitivities $\frac{\partial V_{L,i}}{\partial V_{G,j}}$ have also been used in several SVC control laws (e.g., [15, 96]). Therefore, the matrix $\mathbf{S_V}$ is chosen to be used in the subsequent SVC studies. However, both matrices have been observed to give very similar SVC zoning results.

5

5.2.2. PERFORMANCE EVALUATION

The main goal of SVC is to maintain the voltage at the load buses by controlling the voltage of the pilot buses to their reference values computed by a higher-level optimization program. The regulation of pilot bus voltages is achieved by updating the terminal voltage set points of generators participating in SVC. The secondary objective typically included into SVC is to maximize reactive power reserves by balancing the reactive loading levels of control generators. The reactive loading level is understood as the ratio of reactive power output to the generator reactive power limit.

Across the SVC literature, it is common to evaluate the first SVC objective as the metric of SVC performance. Accordingly, the root mean square error (RMSE) of the load bus voltages is accepted as SVC performance metric:

$$V_{RMSE} = \sqrt{\frac{1}{|\mathcal{L}|}\sum_{i\in\mathcal{L}}(V_i - V_{i,0})^2} \tag{5.8}$$

where $\mathcal{L}$ is the set of load buses, V_i is the voltage magnitude at bus i after voltage control has converged, $V_{i,0}$ is the pre-disturbance reference voltage magnitude at bus i, and V_{RMSE} is the voltage RMSE. Performance metrics similar to (5.8) were used in [96, 176, 177], while mean absolute error (MAE) of the load bus voltages was used in [178]. The minimization of the RMSE metric in (5.8) by SVC stronger penalizes high voltage deviations.

The degree to which the generator reactive loading levels are balanced is often tuned inside the SVC control algorithm (e.g., [15, 96, 179]). A higher emphasis on balancing of reactive loading levels may lead to larger steady-state errors in pilot bus voltages. Therefore, the desired degree of trade-off between the pilot bus voltage regulation and the balancing of generator reactive loading levels should be incorporated into the SVC algorithm (e.g., based on offline studies [15]).

The following two types of voltage errors are introduced for individual system buses to visualize the SVC performance:

$$\Delta V_{i,ad} = |V_{i,ad} - V_{i,0}| \tag{5.9a}$$
$$\Delta V_{i,ac} = |V_{i,ac} - V_{i,0}| \tag{5.9b}$$

where $\Delta V_{i,ad}$ is the absolute voltage deviation at bus i after PVC converges following a disturbance, and $\Delta V_{i,ac}$ is the same voltage deviation after SVC has subsequently converged.

5.2.3. PILOT BUS SELECTION

The used pilot bus selection methodology is similar to the line of work [95, 96], but now the number of pilot buses is chosen from the spectral embedding alignment quality plots. That is, the VCZ configurations featuring high values of the quality function in (4.8) are explored, and each time one pilot bus per zone is selected.

The initial pilot bus selection can be obtained by selecting in each zone the bus with the highest short circuit power [51, 171], the most central bus in terms of electric distance [46], or by using some other heuristics. At the next step, the initial pilot bus selection is improved by changing the pilot buses one at a time and observing the resulting SVC performance. This is similar to the global search method [96], but unlike [96] the pilot bus changes are constrained to their respective control zones, which significantly reduces the search space.

In [96], a specialized power flow algorithm was introduced to quickly and accurately access the SVC performance for multiple operating scenarios and pilot bus configurations. Instead of reproducing this algorithm, the SVC performance is accessed by running the actual coordinated secondary voltage control (CSVC) algorithm proposed in [96] (see Appendix C). As in [52, 96], the SVC performance is evaluated for a single scenario of a positive increase of the reactive power demand of all loads by 25% (more scenarios could be added in practical studies). To ensure the convergence of all pilot bus voltages close to their reference values, the balancing term between the two CSVC objectives has been lowered by the factor of 10 for all case studies, while the other control parameters are as in [96].

5.2.4. CASE STUDIES

This section illustrates the pilot bus selection process using the SVC zoning model of Section 5.2.1 together with the partitioning methodology of Chapter 4 on the example of the IEEE 39 bus test system (the network data is from [180]) and the IEEE 68 bus test system (the network data is from [141]).

IEEE 39 BUS TEST SYSTEM

Test System: The IEEE 39 bus test system [180] consists of 29 load buses and ten generator buses, with the equivalent generator at bus 39 representing the interconnection to an external power system. The generators at buses 30–38 are assumed to participate in SVC, while the equivalent generator is assumed to only maintain the voltage at its terminal bus 39. The one line diagram of this test system together with the SVC zoning results

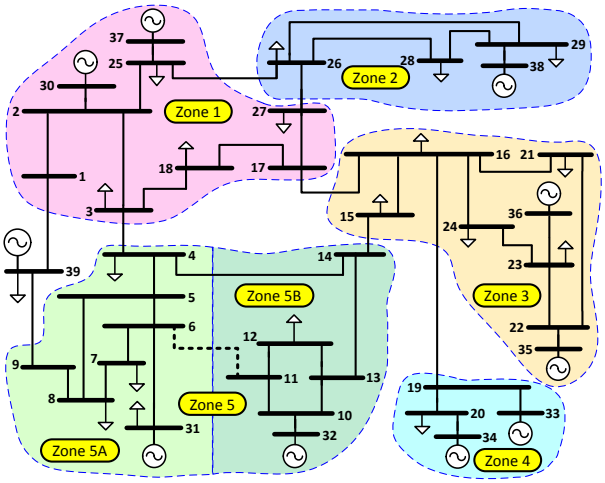


Figure 5.1: Results of AZD of the IEEE 39 bus system with the spectral VCS method

(see below) is shown in Figure 5.1. The model of the IEEE 39 bus test system was taken from MATPOWER, and a few minor differences with the data in [180] were subsequently corrected.

Base Case: To better reflect the bi-objective nature of CSVC, the nominal operating condition of the IEEE 39 bus test system is obtained by running the MATPOWER AC OPF with the generation costs favoring the equal reactive loading levels of all control generators. This step is also necessary to respect the reactive power limits of each generator (as they are given in MATPOWER v6.0b1) after applying load disturbances to the network.

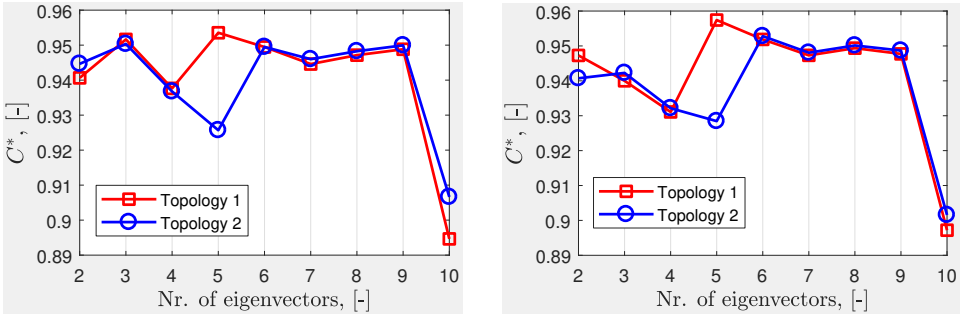
Adaptive Pilot Bus Selection: The benefits of adaptive pilot bus selection are illustrated by simulating the CSVC strategy [96] with various sets of pilot buses for the two topological states of the IEEE 39 bus test network:

1. All elements are in service.

2. Line 6–11 is switched off.

For each of the two operating conditions, the adjacency matrix is constructed both as (5.7) and (5.4), and the alignment quality (4.8) is computed for the number of VCZs ranging from 2 to 10. The results of this process are shown in Figure 5.2 with the optimal number of zones being five for the first topological state and six for the second one. Additionally, it can be noted that the use of both $\mathbf{S_V}$ and $\mathbf{S_Q}$ has resulted in a similar shape of the alignment quality curve.

The two resulting zone divisions are shown in Figure 5.1. As it can be seen, the disconnection of line 6–11 suggests the splitting of the initial Zone 5 into Zone 5A and



(a) Adjacency matrix $\mathbf{S_V}$

(b) Adjacency matrix $\mathbf{S_Q}$

Figure 5.2: Alignment quality (4.8) for the spectral VCS method applied to the IEEE 39 bus test system

Zone 5B. Although the zone border remains the same, the decrease of internal connectedness of Zone 5 effectively leads to its splitting into two clusters, which is well reflected on the alignment quality plot of Figure 5.2. Using the pilot bus selection process of Section 5.2.3, the best found set of pilot buses for the first topological state is {3, 28, 16, 20, 5}, and for the second state it is {3, 28, 16, 20, 5, 12}.

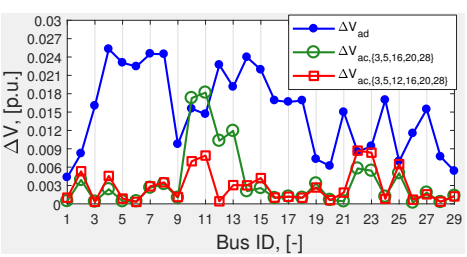
PERFORMANCE EVALUATION

The 25% positive increase of the reactive power demand of all loads results in the total system reactive load increase from 1408.9 MVar to 1807.2 MVar. For this test event, the voltage RMSE (5.8) is summarized in the first three columns of Table 5.1 for the two topological states mentioned above and their corresponding sets of pilots buses. The voltage control performance in the absence of SVC (i.e., with no pilot buses) is also provided as a reference. As it can be seen, the CSVC performance is similar with the two sets of pilot buses for the nominal network topology, but once line 6–11 is switched off, the additional pilot bus 12 starts to create a noticeable difference in the system-wide performance indicator (5.8). The bus-by-bus voltage differences are shown in Figure 5.3a.

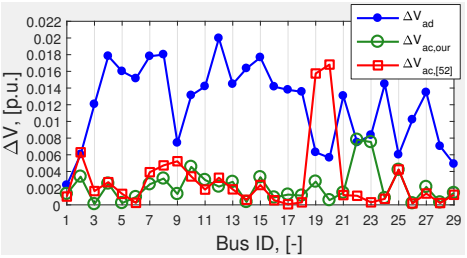
As a reference for comparison, the CSVC algorithm [96] is simulated with the sets of pilot buses originally mentioned in [52] for the same test event of 25% reactive load increase. The results of this case study are also given in Table 5.1 (the last two columns). To comply with the study in [52], the equivalent generator at bus 39 is added to the set of control generators. However, it has been established with the results in Table 5.1 that fixing the voltage at bus 39 instead of choosing a nearby pilot bus yields lower voltage deviations. The results in Table 5.1 demonstrate that the pilot buses obtained with the

Pilots / State	None	{3, 28, 16, 20, 5}	{3, 28, 16, 20, 5, 12}	{1, 3, 6, 24, 28}	{1, 3, 5, 12, 24, 28}
Topology 1	0.0129 p.u.	0.00292 p.u.	0.00292 p.u.	0.00494 p.u.	0.00510 p.u.
Topology 2	0.0164 p.u.	0.00594 p.u.	0.00378 p.u.	0.00664 p.u.	0.00537 p.u.

Table 5.1: Voltage RMSE under AZD for the IEEE 39 bus test system



(a) Topology 2 and pilots {3, 28, 16, 20, 5, 12} vs. {3, 28, 16, 20, 5}



(b) Topology 1 and pilots {3, 28, 16, 20, 5} vs. {1, 3, 6, 24, 28}

Figure 5.3: Bus voltage errors of IEEE 39 bus test system with varying sets of pilot buses and network topologies

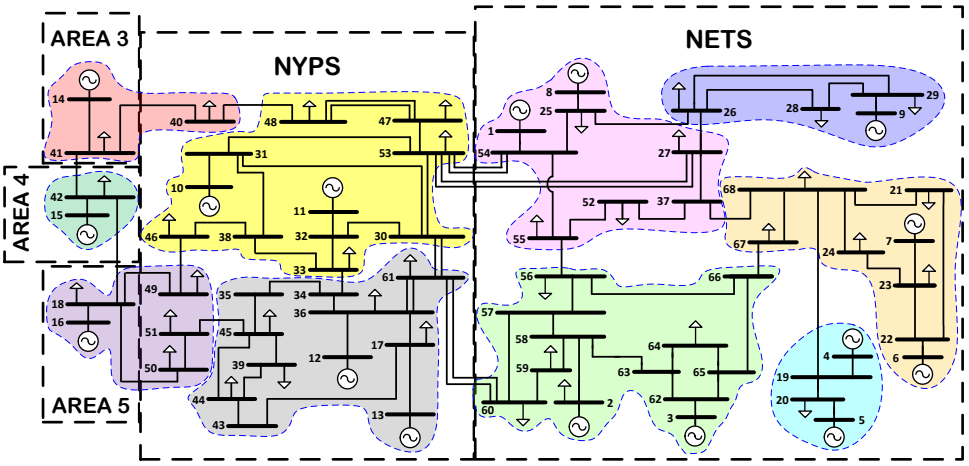


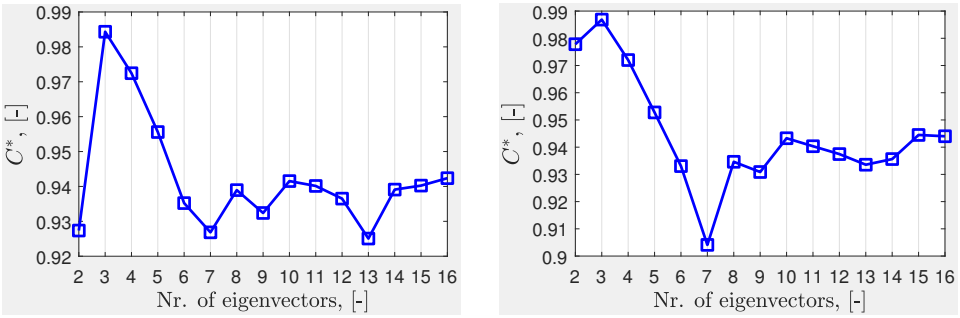
Figure 5.4: Results of AZD of the IEEE 68 bus system with the spectral VCS method

proposed methods show a markedly lower voltage RMSE. The bus-by-bus voltage error comparison is shown in Figure 5.3b.

As it can be noticed, the results in the last two columns of Table 5.1 basically describe the situation with no pilot bus in Zone 4. The addition of bus 20 has resulted in the following performance improvements: the selection {1, 3, 6, 20, 24, 28} for the Topology 1 has dropped the objective (5.8) to 0.00332 p.u., and the selection {1, 3, 5, 12, 20, 24, 28} has resulted in the voltage RMS error of 0.00393 p.u. for the Topology 2. These observations further confirm the ability of the clustering framework in Chapter 4 to infer a good number and location of control zones from the global view of voltage sensitivity relationships between generator and load buses.

IEEE 68 BUS TEST SYSTEM

Test System: The IEEE 68 bus test system [141] consists of 52 load buses and 16 generator buses, with the three equivalent generator buses 14, 15, 16 representing the inter-



(a) Adjacency matrix $\mathbf{S_V}$ (b) Adjacency matrix $\mathbf{S_Q}$

Figure 5.5: Alignment quality (4.8) for the spectral VCS method applied to the IEEE 68 bus test system

connections to external power systems. All the 16 available generators are assumed to participate in SVC. The one line diagram of this test system together with the SVC zoning results (see below) is shown in Figure 5.4. On this diagram, the nominal power system areas are marked with black dashed lines. The NETS area represents the New England test system (i.e., a copy of the IEEE 39 bus test system shown on Figure 5.1 without bus 39), while the NYPS area represents the New York power system and areas 3–5 are the equivalent models of the three other neighboring areas.

Base Case: The nominal power flow scenario of the IEEE 68 bus test system does not result in large violations of generator reactive power limits after applying the 25% reactive load disturbance. This is because the exact generator output power limits are not given in [141], which allows to assume some big numbers instead. Therefore, running the AC OPF to align the pre-disturbance generator reactive power loading levels is unnecessary.

Adaptive Pilot Bus Selection: The pilot buses are selected for the base case scenario similarly to the procedure in Section 5.2.4. The corresponding alignment quality plots are shown in Figure 5.5 for both $\mathbf{S_V}$ and $\mathbf{S_Q}$ used as adjacency matrices.

On the both plots of Figure 5.5, the dominant number of VCZs appears to be three. These zones appear to be the NETS area, area 4, and the NYPS area combined with areas 3 and 5. The voltage variations in area 4 do not propagate further into the system because of the voltage-controlled generators in areas 3 and 5, which explains the formation of a distinct area 4 cluster. Thus, the three zones clustering is meaningful, but it is not very useful for SVC due to the excessive zone sizes. The clustering quality index (4.8) remains relatively high for the 5-zone clustering, which identifies each nominal area in Figure 5.4 as a separate cluster. However, the 5-zone clustering also cannot be accepted because of the excessive sizes of the NETS and NYPS areas. Therefore, the next best number of clusters (i.e., 10 both in Figure 5.5a and 5.5b) is evaluated, and the corresponding zoning results with the matrix $\mathbf{S_V}$ are highlighted in Figure 5.4 by using colors. For this VCZ division, the ten pilot buses have been estimated as {18, 20, 21, 28, 32, 35, 41, 42, 54, 60}.

State \ Pilots	None	{18, 32, 35, 42, 41, 28, 20, 21, 54, 60}
Base case	0.0106 p.u.	0.00393 p.u.

Table 5.2: Voltage RMSE under AZD for the IEEE 68 bus test system

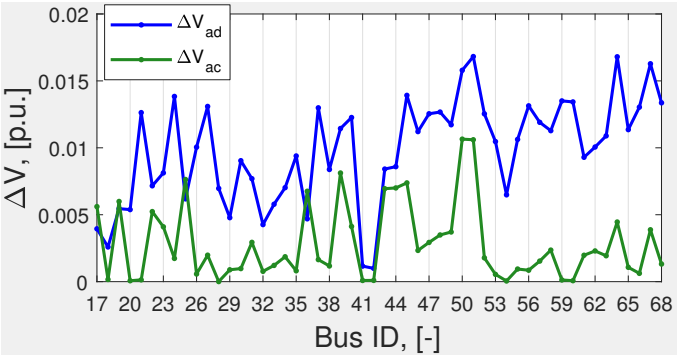


Figure 5.6: Bus voltage errors of IEEE 68 bus test system with the selected pilot buses and with no SVC

Performance evaluation: The 25% positive increase of the reactive power demand of all loads results in the total reactive load increase from 1971.8 MVar to 2654.9 MVar. The resulting voltage RMSE (5.8) for this event is summarized in Table 5.2 both for the case of CSVC not active and CSVC enabled and acting on set of pilot buses selected above. The bus-by-bus voltage error comparison is shown in Figure 5.6.

5.3. GENERATOR SLOW COHERENCY IDENTIFICATION

5.3.1. ANALOGY BETWEEN SLOW COHERENCY AND SPECTRAL CLUSTERING

To apply the ideas from Chapter 4 to the identification of groups of coherent generators, it is first necessary to reveal the close links between generator slow coherency and spectral clustering. Consider again the state matrix in (2.19) that describes the electromechanical mode shapes of a power system:

$$\mathbf{M}^{-1}\mathbf{K} \triangleq \frac{1}{2}\mathbf{H}^{-1}\omega_0\mathbf{K} \tag{5.10}$$

where $M_i = \frac{2H_i}{\omega_0}$ is the scaled inertia constant of machine i and $\mathbf{M} = \text{diag}(M_1, \ldots, M_g)$ is the diagonal matrix of scaled machine inertias.

In (5.10), the entries of the synchronizing torque coefficients matrix $\mathbf{K}$ are computed according to (2.17). In (2.17), the conductances G_{ij} are typically much smaller than the susceptances B_{ij} for high-voltage transmission networks. If the terms with G_{ij} are neglected and there are no phase shifters, the matrix $\mathbf{K}$ becomes symmetric and identical to the *negated* graph Laplacian matrix defined in (2.6a). If the terms G_{ij} are not neglected, their contributions still remain small for the most of practical power networks (e.g., see [34, Chap. 4]), which still keeps $\mathbf{K}$ close to the negated graph Laplacian (2.6a).

At the same time, the inertia matrix $\mathbf{M}$ plays a similar role in (5.10) as the degree matrix $\mathbf{D}$ in (2.6b). As it has been mentioned in Section 2.5, the weighted node degrees on the diagonal of the matrix $\mathbf{D}$ can be interpreted as weights of the respective graph nodes. Similarly, the inertia matrix $\mathbf{M}$ can be seen as the node weights matrix for the graph describing the relationships between generators through synchronizing torque coefficients. The whole process of deriving the spectral relaxation of the Ncut problem summarized in equations (2.8) to (2.12) can be repeated with the matrix $\mathbf{M}$ used instead of the matrix $\mathbf{D}$. For example, the resulting *generic normalized cut* is similar to (2.8):

$$\text{minimize} \quad \text{Ncut}_{\text{M}}(\widetilde{\mathbf{X}}) = \frac{1}{k}\sum_{j=1}^{k}\frac{\widetilde{\boldsymbol{X}}_j^T\left(-\mathbf{K}^{\mathbf{S}}\right)\widetilde{\boldsymbol{X}}_j}{\widetilde{\boldsymbol{X}}_j^T\mathbf{M}\widetilde{\boldsymbol{X}}_j} \tag{5.11a}$$

$$\text{subject to:} \quad \widetilde{\mathbf{X}} \in \{0,1\}^{g\times k},\ \widetilde{\mathbf{X}}\mathbf{1}_{k\times 1} = \mathbf{1}_{g\times 1} \tag{5.11b}$$

where $\widetilde{\mathbf{X}} = [\widetilde{\boldsymbol{X}}_1,\ldots,\widetilde{\boldsymbol{X}}_k]$ is the partition indicator matrix of generator groups $\mathcal{C}_1,\ldots,\mathcal{C}_k$, $\widetilde{\boldsymbol{X}}_j$ are the indicator vectors of the individual groups, and $\mathbf{K}^{\mathbf{S}}$ is a symmetric version of the synchronizing torque coefficients matrix $\mathbf{K}$. The optimized quantity in (5.11) is called generic normalized cut due to the generic node weights in the matrix $\mathbf{M}$ being present in the denominator instead of the degree matrix $\mathbf{D}$ as in (2.8). The symmetry of the matrix $\mathbf{K}$ is necessary for the clean derivation of the spectral relaxation in (5.11) through (2.8)–(2.12) and for the evaluation of the Ncut_{M} criterion (5.11). The matrix $\mathbf{K}^{\mathbf{S}}$ can be defined by neglecting the conductances G_{ij} in the Kron-reduced admittance matrix in (2.17) (this matrix is specifically referred to as $\mathbf{K}^{\mathbf{B}}$), or by symmetrizing the original matrix $\mathbf{K}$ with a matrix symmetrization method. Coincidentally, if the real and imaginary parts of the Kron-reduced admittance matrix in (2.17) are both symmetric, which is valid for power networks without phase shifters, symmetrizing $\mathbf{K}$ by averaging the elements above and below the main diagonal as $K^S_{ij} = (K_{ij} + K_{ji})/2$ and then adjusting the diagonal elements as $K^S_{ii} = -\sum_{j=1,\ j\neq i}^{g} K^S_{ij}$ will result in $\mathbf{K}^{\mathbf{S}} = \mathbf{K}^{\mathbf{B}}$.

Analogously to (2.23), it is possible to introduce the *generic expansion ratio* (or *generic expansion*) of a single generator group $\mathcal{C}$ as the contribution of this group to the overall generic normalized cut (5.11):

$$\phi_M(\mathcal{C}) = \frac{\widetilde{\boldsymbol{X}}_{\mathcal{C}}^T\mathbf{K}^{\mathbf{S}}\widetilde{\boldsymbol{X}}_{\mathcal{C}}}{\widetilde{\boldsymbol{X}}_{\mathcal{C}}^T\mathbf{M}\widetilde{\boldsymbol{X}}_{\mathcal{C}}} = \frac{\sum_{i\in\mathcal{C},\,j\in\mathcal{G}\setminus\mathcal{C}} K^S_{ij}}{\sum_{i\in\mathcal{C}} M_i} \tag{5.12}$$

where $\widetilde{\boldsymbol{X}}_{\mathcal{C}}$ is the binary indicator vector of the generator group $\mathcal{C}$.

From Section 2.5 and the references therein, the spectral relaxation of (5.11) expressed with the eigenvectors of $\mathbf{M}^{-1}\mathbf{K}^{\mathbf{S}}$ is given by:

$$\text{minimize} \quad \text{Ncut}^{\text{SR}}_{\text{M}}(\mathbf{V}) = \frac{1}{k}\,\text{tr}\left(\mathbf{V}^T\left(-\mathbf{K}^{\mathbf{S}}\right)\mathbf{V}\right) \tag{5.13a}$$

$$\text{subject to}: \quad \mathbf{V}^T\mathbf{M}\mathbf{V} = \mathbf{I}_{\mathbf{k}} \tag{5.13b}$$

and a relaxation analogous to (2.11) can be devised by using the closest to zero eigenpairs of the *symmetric* matrix $\mathbf{M}^{-1/2}\mathbf{K}^{\mathbf{S}}\mathbf{M}^{-1/2}$ that is similar to $\mathbf{M}^{-1}\mathbf{K}^{\mathbf{S}}$. In (5.13), constraint (5.13b) states that the eigenvectors of $\mathbf{M}^{-1}\mathbf{K}^{\mathbf{S}}$ should be M-orthogonal to repre-

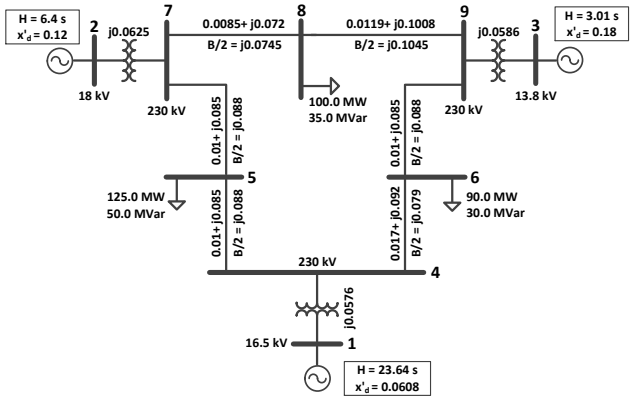


Figure 5.7: Electromechanical model data of the IEEE 9 bus system as given in [34] (100 MVA base power)

5

sent the optimal solution to the relaxation (5.13) of (5.11). Systems of differential equations of the form (2.19) are common in theory of mechanical vibrations [181], where the matrix $\mathbf{M}$ is the mass matrix, matrix $\mathbf{K^S}$ is the (symmetrical) stiffness matrix, and the M-orthogonal eigenvectors $\mathbf{V}$ are known as *mass-normalized* mode shapes. The eigenvectors of $\mathbf{M}^{-1}\mathbf{K^S}$ can be mass-normalized to satisfy (5.13b) as follows:

$$\boldsymbol{Z_j} = \frac{\boldsymbol{V_j}}{\boldsymbol{V_j}^T \mathbf{M} \boldsymbol{V_j}}, \quad j = 1, \ldots, g \tag{5.14}$$

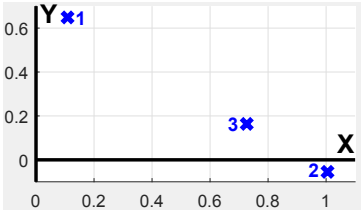
where $\boldsymbol{V_j}$ is the j eigenvector of $\mathbf{M}^{-1}\mathbf{K^S}$ (arbitrarily scaled), and $\boldsymbol{Z_j}$ is the corresponding mass-normalized eigenvector.

For the classic slow coherency grouping algorithm proposed in [34, Chapter 5], [33, 182], the scaling of eigenvectors was not elaborated except mentioning that, once group reference machines are selected, the grouping does not depend on the eigenvector scaling. For another well-known tolerance-based generator coherency algorithm [35, 121], it was recommended to normalize each eigenvector of $\mathbf{M}^{-1}\mathbf{K^S}$ to length one, which is similar to the popular feature scaling approach in machine learning.

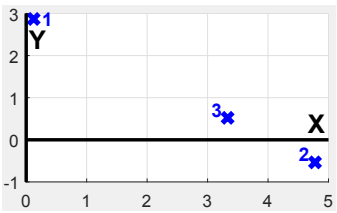
To illustrate the advantage of using mass-normalized electromechanical eigenvectors, consider the slow coherency grouping of the generators of the IEEE 9 bus test system from [34, Chap. 4] shown in Figure 5.7. The unit-length normalized and mass-normalized eigenvectors of the matrix $\mathbf{M}^{-1}\mathbf{K}$ corresponding its three eigenvalues {0, -75.54, -179.15} are listed below:

$$\mathbf{V} = \begin{bmatrix} 0.57735 & 0.31499 & -0.04004 \\ 0.57735 & -0.82430 & -0.29591 \\ 0.57735 & -0.47043 & 0.95438 \end{bmatrix} \qquad \mathbf{Z} = \begin{bmatrix} 2.3882 & 1.5940 & -0.3008 \\ 2.3882 & -4.1715 & -2.2230 \\ 2.3882 & -2.3807 & 7.1697 \end{bmatrix} \tag{5.15}$$

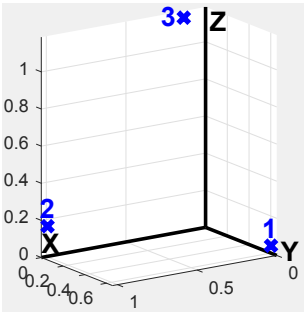
where the eigenvector columns in $\mathbf{V}$ are unit-length normalized, and the eigenvector columns in $\mathbf{Z}$ are mass-normalized. As the next step, the eigenvector alignment algo-



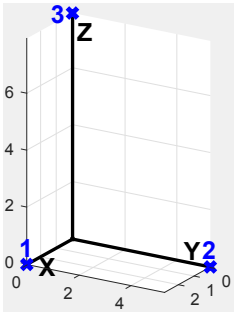
(a) First 2 modes, unity-length normalized



(b) First 2 modes, mass-normalized



(c) First 3 modes, unity-length normalized



(d) First 3 modes, mass-normalized

Figure 5.8: Aligned eigenvectors of the electromechanical model of the IEEE 9 bus system

rithm from Section 4.3.3 is tested on the both eigenvector matrices in (5.15). The results of aligning the first two and three columns of **V** and **Z** are shown in Figure 5.8.

Figure 5.8 demonstrates the excellent alignment of the mass-normalized electromechanical eigenvectors with the canonical coordinate axes. This is despite the fact that the eigenvectors in (5.15) are computed for the original non-symmetric matrix **K**. The spectral embedding alignment algorithm from Chapter 4 is used at this stage to compute from (5.15) the orthogonally-transformed spectral embeddings in Figure 5.8. Finally, scaling eigenvectors to the unit length may be well-justified for the tolerance-based coherency algorithm [121], as this algorithm does not rely on the orthogonal structure of eigenvectors and the overall clustering structure is well-preserved with unity-length normalized eigenvectors. However, the additional information in the orthogonal structure of electromechanical eigenvectors may provide some new possibilities for the design of generator coherency identification algorithms.

5.3.2. SLOW COHERENCY GROUPING ALGORITHM

The demonstrated results on the alignment of *mass-normalized* electromechanical eigenvectors to the canonical coordinate system motivate the development of a new generator coherency identification algorithm. The proposed algorithm combines the ideas about Laplacian eigenvector alignment from Chapter 4 with the classic ideas about generator coherency from [33, 34, 182]. Similarly to the k-way partitioning algorithm of

Section 4.4, the developed coherency algorithm includes the alignment cost minimization stage and the minimization of normalized cut.

ALIGNMENT COST MINIMIZATION FOR GENERATOR COHERENCY

The eigenvector alignment cost minimization stage allows to select sets of slowest eigenvectors of the matrix $\mathbf{M}^{-1}\mathbf{K}$ that are most representative at describing the electromechanical areas of a power network. The basic procedure is as in Section 4.3: compute the $k_{\max}$ slowest eigenvectors of $\mathbf{M}^{-1}\mathbf{K}$ and incrementally align the first k of these eigenvectors with the canonical coordinate system for k running from 2 to $k_{\max}$. However, there are some differences compared to Chapter 4 that are described below.

First, using the slow eigenvectors of $\mathbf{M}^{-1}\mathbf{K}$ tends to produce good results despite the matrix $\mathbf{K}$ usually being non-symmetric. However, it may be advised to test the spectrum of the original $\mathbf{M}^{-1}\mathbf{K}$ against the spectrum of its counterpart involving a symmetric version of $\mathbf{K}$ to check how close is the matrix $\mathbf{K}$ to being symmetric (e.g., see [34, Chap. 4]).

Second, if scaled generator inertias are used as graph node weights, using an analog of the matrix $\mathbf{W_n}$ (2.13b) to speed up the computation of eigenvalues will not work, as the eigenvectors of $\mathbf{W_n}$ match those of $\mathbf{L_n}$ (2.6b) and $\mathbf{L_{rw}}$ (2.6c) only if the diagonal matrix of graph node weights is the degree matrix $\mathbf{D}$ (2.2).[2]

Finally, the row normalization procedure (2.14) is not applied to the mass-normalized eigenvector matrix $\mathbf{Z} \in \mathbb{R}^{g\times k}$ that is used to identify the coherent areas. According to [33, 182], it is beneficial to choose generators with *large* and *linearly-independent* rows in the electromechanical eigenvector matrix as *reference generators* of the coherent areas. Thus, normalizing the length of each row of $\mathbf{Z}$ to one would exclude some information that is valuable for the generator coherency analysis. In addition, normalizing the rows of $\mathbf{Z}$ to length one results in columns of the resulting matrix $\mathbf{X}$ no longer being valid eigenvectors of $\mathbf{M}^{-1}\mathbf{K}$.

As the rows of $\mathbf{Z}$ are not normalized to unit length, the alignment quality metrics in (4.2) and (4.8) are not directly applicable. Therefore, the cost in (4.5) is used to assess the alignment of the orthogonally transformed matrix $\mathbf{Z}$ with the canonical coordinate system. Although it is possible to directly rotate the original eigenvectors in $\mathbf{Z}$ towards the canonical coordinate system by using the GD-based algorithm of Section 4.3.1 and Appendix B, the actual performance of this approach has been observed to be inferior compared to the techniques from Sections 4.3.1–4.3.3 that involve the row normalization of $\mathbf{Z}$ as in (2.14). That is, the row-normalized spectral embedding $\mathbf{X} \in \mathbb{R}^{g\times k}$ is aligned with the standard basis by using the techniques from Sections 4.3.1–4.3.3, and the computed orthogonal transformation $\mathbf{R}^*$ is applied to the initial eigenvectors in $\mathbf{Z} \in \mathbb{R}^{g\times k}$. The GD-based algorithm to minimize (4.5) is used afterwards to refine the alignment.

GRAPH PARTITIONING FOR GENERATOR COHERENCY

Similarly to the classic slow coherency approach [33, 34, 182], the proposed generator grouping algorithm through normalized cut minimization seeks to find k generator

[2] Section 2.5 describes how the eigenvalues of $\mathbf{W_n} = \mathbf{I} - \mathbf{L_n}$ are shifted by 1 from those of $-\mathbf{L_n}$ with the corresponding eigenvectors being equal. This correspondence between the eigenpairs is only valid if the two matrices differ by a scalar multiple of the identity matrix – the property that is lost if the matrix $\mathbf{M}$ is used in (2.6b), (2.6c), (2.13a), (2.13b) instead of the degree matrix $\mathbf{D}$.

Algorithm 5.1 Estimation of Generator Group Cores

Input: Aligned spectral embedding $\mathbf{Z}^*_{g\times k}$, matrix $\mathbf{K^S}$, matrix $\mathbf{M}$

$\beta \leftarrow \sqrt{2}/2$ // Set the eigenvector threshold β to the lowest value
Reorder the columns of $\mathbf{Z}^*$ in the descending order of the maximum absolute elements of each column.
$\mathbf{X}^* \leftarrow \text{diag}(\text{Diag}(\mathbf{Z}^*\mathbf{Z}^{*T}))^{-\frac{1}{2}}\mathbf{Z}^*$ // Row-normalized spectral embedding corresponding to $\mathbf{Z}^*$
for $j = 1$ to k **do**
    $\boldsymbol{Z} \leftarrow \mathbf{Z}^*[1,\ldots,g;j]$ // j^{th} column of $\mathbf{Z}^*$
    $\boldsymbol{X} \leftarrow \mathbf{X}^*[1,\ldots,g;j]$ // j^{th} column of $\mathbf{X}^*$
    $\boldsymbol{ord} \leftarrow$ Descending order of entries in $\boldsymbol{Z}$
    $l \leftarrow 0$
    for $i \leftarrow 1,\ldots,g$ **do**
        if $\boldsymbol{X}[\boldsymbol{ord}[i]] > \beta$ **then**
            $l \leftarrow l+1$
            $\boldsymbol{core}[l] \leftarrow \boldsymbol{ord}[i]$
            $\boldsymbol{phi}[l] \leftarrow \phi_M(\mathbf{K^S},\mathbf{M},\boldsymbol{core})$ // (5.12), (5.16)
        end if
    end for
    $l^* \leftarrow \text{argmin}\,\boldsymbol{phi}$
    $\mathcal{CC}_j \leftarrow \{\boldsymbol{core}[1,\ldots,l^*]\}$
    $\mathbf{Z}^*[\mathcal{CC}_j;1,\ldots,k] = 0$
    $\mathbf{X}^*[\mathcal{CC}_j;1,\ldots,k] = 0$
end for

Output: Generator group cores $\mathcal{CC}_1,\ldots,\mathcal{CC}_k$

groups for a given matrix $\mathbf{Z}^*$ containing the slowest k mass-normalized aligned eigenvectors of $\mathbf{M}^{-1}\mathbf{K}$ as its columns. The methodology to estimate generator group cores from aligned electromechanical mode shapes is summarized as Algorithm 5.1, and it is based on Section 4.4 with some modifications (cf. Algorithm 4.2).

New generators are added to a group core in the descending order of magnitudes of the corresponding column of $\mathbf{Z}^*$ instead of $\mathbf{X}^*_{g\times k}$. In this way, the largest rows of $\mathbf{Z}^*$ that best indicate the reference machine of the current generator group [33, 182] are always considered first. However, without row normalization, the entries' magnitudes in the columns of $\mathbf{Z}^*$ may vary considerably (e.g., see Figure 5.8d), which makes it difficult to set a fixed lower threshold for considering each column of $\mathbf{Z}^*$ independently. To solve this problem, the columns of the row-normalized matrix $\mathbf{X}^*$ are used in combination with a fixed threshold value β similarly to Algorithm 4.2, except that the order of entries in the current column of $\mathbf{X}^*$ is determined by the descending order of magnitudes of the corresponding column of $\mathbf{Z}^*$. To illustrate the interplay of the columns of $\mathbf{Z}^*$ and $\mathbf{X}^*$, Figure 5.9 shows the generator grouping process for the NPCC 48 machine power system [34, 35] into three areas by using the three slowest eigenvectors of (5.10) (axes-aligned and mass-normalized). A copy of the geographical diagram of the NPCC 48 machine power system is shown in Section 5.3.5.

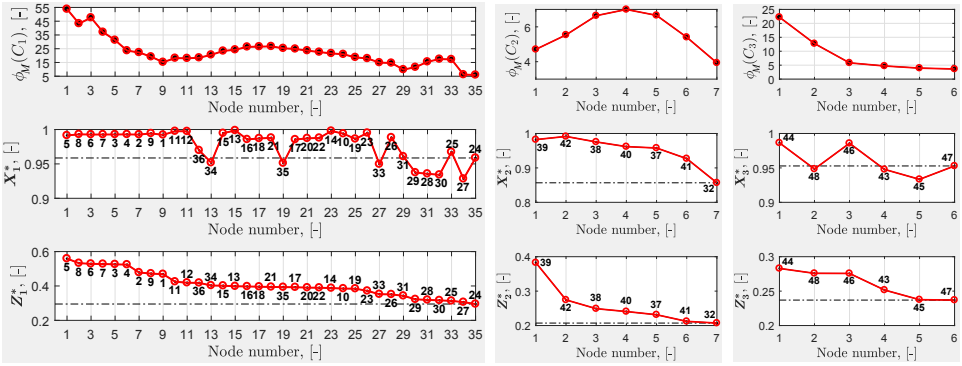


Figure 5.9: Estimation of three slowly coherent generator groups of the NPCC 48 machine test system [34] by using Algorithm 5.1. Each column of $\mathbf{Z}^*$ is sorted in descending order that defines the ordering of the corresponding columns of $\mathbf{X}^*$. The generic expansion ϕ_M is computed using the symmetric matrix $\mathbf{K^B}$.

5

Another issue that arises due to possible considerable differences in magnitudes of entries of various columns of $\mathbf{Z}^*$ is the possibility to have the same row of $\mathbf{Z}^*$ hosting multiple column maxima. In such situations, the same generator can be assigned to multiple groups unless some special measures are taken. For this reason, lines 2 and 18–19 are included into Algorithm 5.1. The permutation of the columns of $\mathbf{Z}^*$ at line 2 establishes the priority order for the columns of $\mathbf{Z}^*$ so that the prospective generator groups featuring largest entries in their respective columns are considered first. Additionally, lines 18–19 ensure that no row of $\mathbf{Z}^*$ is selected more than once during the grouping process.

Unlike in Algorithm 4.2, it is suggested to set the threshold β to the lowest reliable value of 0.707 when clustering the columns in $\mathbf{Z}^*$. Moreover, it was observed that setting β to values somewhat lower than 0.707 may sometimes improve the $\mathrm{Ncut_M}$ values and slow coherency results (in terms of the criteria discussed in Section 5.3.3) when the eigenvector alignment cost is relatively high. As β is set to the lowest reliable value, the gains achievable by the cluster core refinement described in Section 4.4 diminish. Consequently, the cluster core refinement stage is absent in Algorithm 5.1, which also simplifies the algorithm. Finally, the minimum cluster size requirement looses its meaning in the slow coherency context (i.e., if a single large equivalent generator constitutes a separate group, it must be treated as such).

An execution of Algorithm 5.1 may not classify every machine to a group core. The *remaining machines* that do not belong to any group core form a set $\mathcal{R}$. These machines are similar to *loosely coherent machines* from literature [34, 35]. Because the involved matrices $\mathbf{K}$ or $\mathbf{K^B}$ correspond to a complete graph, the remaining machines can be directly assigned in any order to any group. Below we propose a greedy algorithm to accomplish this that usually produces similar or better results as the min-cut based recursive bisection in Section 4.4:

1. Evaluate the generic expansions (5.12) of each of the machine group cores $\mathcal{CC}_1,\ldots,\mathcal{CC}_k$ obtained from Algorithm 5.1 as $\phi_M(\mathcal{CC}_1),\ldots,\phi_M(\mathcal{CC}_k)$.

2. Attempt to move each remaining machine $j \in \mathcal{R}$ to every group core and evaluate the signed differences in generic expansions $\Delta\phi_M(\mathcal{CC}_l \cup j) = \phi_M(\mathcal{CC}_l) - \phi_M(\mathcal{CC}_l \cup j),\ l = 1,\ldots,k,\ j \in \mathcal{R}$ resulting from these moves.
3. Identify the largest signed difference $\Delta\phi_M(\mathcal{CC}_{l*} \cup j^*)$, where $\mathcal{CC}_{l*}$ and j^* are respectively the group core and the remaining machine involved in the move.
4. Perform the updates: $\phi_M(\mathcal{CC}_{l*}) \leftarrow \phi_M(\mathcal{CC}_{l*} \cup j^*), \mathcal{CC}_{l*} \leftarrow \mathcal{CC}_{l*} \cup j^*, \mathcal{R} \leftarrow \mathcal{R} \setminus j^*$.
5. Repeat steps 2, 3, 4 until all remaining machines are assigned to a group core.

In the above algorithm, the incremental updates of group cores expansions at step 2 can be implemented efficiently by using the formula for ϕ_M analogous to (4.9):

$$\phi_M(\mathcal{C}) = \frac{\mathrm{vol}(\mathcal{C}) - \mathrm{links}(\mathcal{C},\mathcal{C})}{\sum_{i\in\mathcal{C}} M_i} \tag{5.16}$$

where $\mathrm{vol}(\mathcal{C})$ and $\mathrm{links}(\mathcal{C},\mathcal{C})$ are based on (2.1), (2.4) and (2.3) with the underlying adjacency matrix being $\mathbf{W^S} = \mathbf{K^S} - \mathrm{diag}(\mathrm{Diag}(\mathbf{K^S}))$. To compute the change of $\phi_M(\mathcal{C})$ due to moving a single machine to or from the group $\mathcal{C}$, the quantities $\mathrm{vol}(\mathcal{C})$, $\mathrm{links}(\mathcal{C},\mathcal{C})$ and the sum $\sum_{i\in\mathcal{C}} M_i$ need to be updated accordingly. Updating $\mathrm{vol}(\mathcal{C})$ and $\sum_{i\in\mathcal{C}} M_i$ involves a simple addition or subtraction, as the weighted degrees d_i can be precomputed for a given $\mathbf{W^S}$ and the scaled machine inertias M_i belong to the problem definition in (5.10). Updating $\mathrm{links}(\mathcal{C},\mathcal{C})$ involves adding or subtracting the term $2\sum_{i\in\mathcal{C}} W^S_{ij}$, where j is the machine moved to or from $\mathcal{C}$. The above update scheme is also used in Algorithm 5.1 at line 13.

The mass-normalized eigenvectors in $\mathbf{Z}$ represent an optimal solution to a *continuous relaxation* (5.13) of the NP-complete problem in (5.11). This implies a possibility of suboptimal groupings by Algorithm 5.1, especially if the alignment cost in (4.5) is high. The greedy machine assignments described above may also lead the solution away from the global optimum of (5.11). Because of these complications, the obtained solutions may noticeably benefit from graph cut refinement [C2]. The following algorithm similar to greedy assignment algorithm described above has been successfully applied to improve the $\mathrm{Ncut_M}$ values:

1. Evaluate the generic expansions (5.12) of the input machine groups $\mathcal{C}_1,\ldots,\mathcal{C}_k$ as $\phi_M(\mathcal{C}_1),\ldots,\phi_M(\mathcal{C}_k)$.
2. Attempt to move each machine $j \in \mathcal{G}$ from its own group $\mathcal{C}_f$ to every other group $\mathcal{C}_t$ unless $\mathcal{C}_f$ is a group containing a single machine. Evaluate the signed differences $\Delta\phi_M(\mathcal{C}_t \cup j) = \phi_M(\mathcal{C}_t) - \phi_M(\mathcal{C}_t \cup j)$ and $\Delta\phi_M(\mathcal{C}_f \setminus j) = \phi_M(\mathcal{C}_f) - \phi_M(\mathcal{C}_f \setminus j)$ for $j \in \mathcal{G},\ (\mathcal{C}_f \ni j) \wedge (|\mathcal{C}_f| > 1),\ \mathcal{C}_t \in \{\mathcal{C}_1,\ldots,\mathcal{C}_k\} \setminus \mathcal{C}_f$.
3. Identify the largest signed difference $\Delta\phi_M(\mathcal{C}_{t*}, j^*) = \Delta\phi_M(\mathcal{C}_{t*} \cup j^*) + \Delta\phi_M(\mathcal{C}_{f*} \setminus j^*)$, where $\mathcal{C}_{t*}$ and j^* are respectively the receiving group and the machine to be moved from the sending group $\mathcal{C}_{f*}$.
4. If $\Delta\phi_M(\mathcal{C}_{t*}, j^*) > 0$, perform the updates: $\phi_M(\mathcal{C}_{t*}) \leftarrow \phi_M(\mathcal{C}_{t*} \cup j^*),\ \phi_M(\mathcal{C}_{f*}) \leftarrow \phi_M(\mathcal{C}_{f*} \setminus j^*), \mathcal{C}_{t*} \leftarrow \mathcal{C}_{t*} \cup j^*, \mathcal{C}_{f*} \leftarrow \mathcal{C}_{f*} \setminus j^*$.
5. Repeat steps 2, 3, 4 until $\Delta\phi_M(\mathcal{C}_{t*}, j^*) \le 0$.

The above algorithm also exploits the possibility to move any machine to any group and achieves a high efficiency by using the update scheme based on (5.16) at step 2. The full potential of the Ncut refinement algorithm can be realized by running it multiple times with different initial machine groupings. Meaningful initial machine groupings can be generated based on the output results of Algorithm 5.1 (i.e., the machine group cores and the set of remaining machines $\mathcal{R}$). In Section 5.3.5, the initial machine groupings were obtained in the following three ways:

- A run of Algorithm 5.1 followed by the greedy assignment of the remaining machines.
- A run of Algorithm 5.1 followed by a random group assignment of the remaining machines. In Section 5.3.5, 15 random initializations of this type were used.
- Rounding of the aligned matrix $\mathbf{Z}^*$. Each machine's group is defined by the column index of the maximum value of the row of $\mathbf{Z}^*$ corresponding to that machine.

If the Ncut refinement algorithm is not applied, the last two grouping types usually do not produce good results, so their value is entirely in providing meaningful starting points to the Ncut refinement process.

5.3.3. EVALUATION OF GENERATOR COHERENCY RESULTS

According to multiple references (e.g., [33, 34, 35, 121, 182]), it is possible to access the accuracy of a slow coherency generator grouping through the following approaches:

1. By comparing the slow eigenvalues of the original model (2.19) with the eigenvalues of a reduced linear model based on the slow coherency generator grouping.
2. By comparing the time domain disturbance responses of the original and reduced nonlinear model obtained from the slow coherency generator grouping.

For the first approach, the linear inertial aggregate model that is briefly described in Appendix D is going to be used. The eigenvalue approximation accuracy by the inertial aggregate model can be improved through adding the first-order correction terms, which results in the *slow coherency aggregate model* [34, 35]. However, the goal of the grouping evaluation procedure is not to obtain a low error in slow eigenvalues, but to see which machine grouping results in lower eigenvalue errors regardless of the used aggregation method. That is, the slow coherency aggregate model is able to improve errors in slow electromechanical eigenvalues, but not the grouping itself, and an inaccurate grouping will still produce higher eigenvalue errors as compared to a more accurate one. Therefore, using only the inertial aggregate model is sufficient for comparing errors in the first k slow eigenvalues between various machine groupings.

To evaluate the generator grouping accuracy with the first approach, the mean percentage eigenvalue error is used as the total metric:

$$\overline{\delta\lambda} = \frac{1}{k}\sum_{i=1}^{k}\frac{|\lambda_{a,i}-\lambda_{f,i}|}{|\lambda_{f,i}|}\cdot 100 \tag{5.17}$$

where $\lambda_{f,i}$ is the i slowest eigenvalue of the full electromechanical model (2.19) and $\lambda_{a,i}$ is the eigenvalue of the reduced electromechanical model closest to $\lambda_{f,i}$. Additionally,

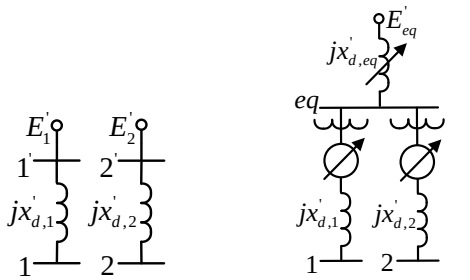
for certain generator groupings it is possible to separately consider each slow eigenvalue of the original and reduced electromechanical models.

To evaluate the generator grouping accuracy with the second approach, the nonlinear model reduction should be used (see [35, 183]). Unlike the linear inertial aggregate model described in Appendix D, nonlinear model reduction produces power system models that consist of separate power system elements such as generators, loads and transmission lines, which enables their use in standard time-domain simulation programs (e.g., for DSA). According to [35, 183], the inertial and slow coherency aggregations show comparable results in the context of nonlinear model reduction for time-domain simulations, with the inertial aggregation algorithm being simpler and more intuitive. Therefore, reduced nonlinear power system models are produced by the inertial aggregation algorithm [183] that is available in PST as function `i_agg()`.

The comparison of disturbance responses between the reduced and full power system models also follows the lines of [35, 183]. First, synchronous machines are grouped based on slow coherency. For a given grouping, one or multiple coherent groups corresponding to a study area of interest are selected. While the study area is retained as it is, the remaining coherent groups are reduced to single equivalent generators by using the nonlinear generator aggregation algorithm of choice. Finally, some disturbances are simulated in the study area both with the full and reduced system models, and the resulting responses are compared. The deviation of reduced-model signals from the reference provided by the simulation of the full model can be quantified by RMSE or MAE integrals over a time period T [183]. However, in many cases a simple visual inspection of time responses is enough to recognize the more successful machine grouping.

5.3.4. IMPROVED INERTIAL GENERATOR AGGREGATION ALGORITHM

An important step of the nonlinear model reduction that was briefly discussed in Section 5.3.3 is the generator aggregation process that consists in replacing a group of coherent generators by a single equivalent machine through a special algorithm. A known drawback of generator aggregation is the *stiffening effect* that manifests itself in the increased electromechanical mode frequencies of the reduced model compared to the original one. An important motivation behind the inertial generator aggregation algorithm men-



(a) Before aggregation (b) After aggregation

Figure 5.10: Improved inertial generator aggregation algorithm

Algorithm 5.2 Improved Inertial Generator Aggregation

Input: Power system model, coherent groups $\mathcal{C}_1, \ldots, \mathcal{C}_k$.
for $l = 1$ to k **do**
  $\widehat{x}'_{d,eq} \leftarrow 20\left(\sum_{i\in\mathcal{C}_l} 1/x'_{d,i}\right)^{-1}$
  $\boldsymbol{x} \leftarrow$ Range of numbers from 10^{-6} to $\widehat{x}'_{d,eq}$.
  for $i = 1$ to $|x|$ **do**
    Aggregate group $\mathcal{C}_l$ with the basic method [183].
    $x'_{d,eq} \leftarrow \boldsymbol{x}[i]$
    Compute the slow modes and store errors (5.17).
  end for
  Choose $x'_{d,eq}$ for $\mathcal{C}_l$ as $\boldsymbol{x}[i]$ yielding smallest error (5.17).
end for
Output: Reduced power system model

5

tioned in Section 5.3.3 is the reduction of the stiffening effect, which is achieved by aggregating the coherent generators at their internal nodes instead of the more conventional terminal bus aggregation. Merging together the internal generator nodes of each coherent group as shown in [183] and linearizing the resulting reduced system produces the same model as the linear inertial aggregate model (see Appendix D). As it will become clear from Section 5.3.5, the linear inertial aggregate model still results in considerable eigenvalue errors (5.17), which implies that the achieved mitigation of the stiffening effect is incomplete.

As the persisting stiffening effect of the inertial aggregation algorithm may often lead to unsatisfactory nonlinear reduced models, a simple yet effective improvement is proposed. In Figure 5.10, the initial and final stages of the inertial aggregation algorithm are illustrated for two generators represented by their classical models. Going from Figure 5.10a to Figure 5.10b involves aggregation of nodes $1'$ and $2'$ into node eq using the Zhukov bus aggregation method [184]. This includes adding ideal transformers and phase shifters in series with reactances $x'_{d,1}$ and $x'_{d,2}$ to preserve the initial power flow. The dynamic parameters of the equivalent generator are derived from the sum of swing equations of generators belonging to a coherent group $\mathcal{C}$ by assuming their incremental speeds and rotor angles to be the same (the coherency condition):

$$H_{eq} = \sum_{i\in\mathcal{C}} H_i,\ D_{eq} = \sum_{i\in\mathcal{C}} D_i \tag{5.18}$$

where H_{eq} and D_{eq} are the equivalent inertia and damping constants.

In the original inertial aggregation algorithm [183], the effective reactance behind bus eq is zero (i.e., $x'_{d,eq} = 0$), which strictly corresponds to the aggregation of generator internal buses. However, we have noticed that it is possible to decrease the electromechanical frequencies of the reduced model by increasing the reactance $x'_{d,eq}$ from a value close to zero to some meaningful upper limit $\widehat{x}'_{d,eq}$. The improved inertial aggregation algorithm is summarized as Algorithm 5.2. In Algorithm 5.2, $\widehat{x}'_{d,eq}$ is set to 20

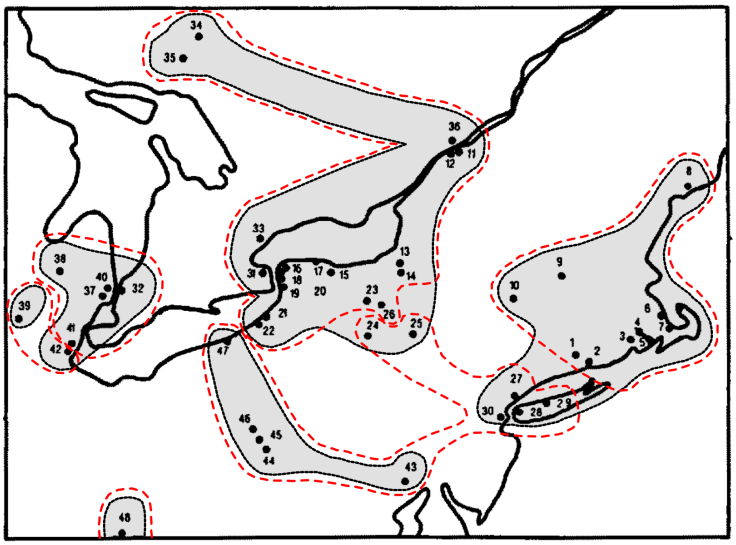


Figure 5.11: Geographical diagram of the NPCC 48 machine test system (a copy from [34]). The 6-area groupings are by the algorithm of Section 5.3.2 (shown in grey) and the classic slow coherency grouping algorithm from [33, 34] (shown by red dashed lines). $\mathbf{M}^{-1}\mathbf{K}$ serves as the electromechanical model.

times $x'_{d,eq}$ of the Podmore algorithm, which is chosen empirically for studies on the NPCC 48-machine test system.

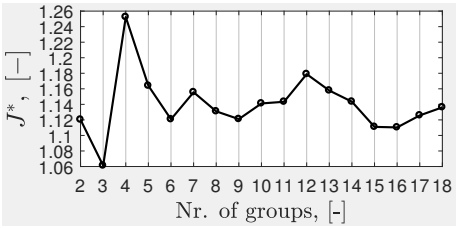
Algorithm 5.2 preserves the advantages of the baseline approach [183] such as conceptual simplicity and possibility to independently aggregate each generator group. As shown in Section 5.3.5, Algorithm 5.2 is significantly more accurate, thus trading execution speed for accuracy. However, if execution speed is highly important, Algorithm 5.2 allows for massive parallelization. Clearly, both basic inertial aggregation and Algorithm 5.2 are only valid for generators represented by the classical model. However, representing external remote generators by the simplified 2^{nd} order model is well-accepted and often used in practice (e.g., in the DYNRED software [185]).

5.3.5. GENERATOR COHERENCY CASE STUDIES

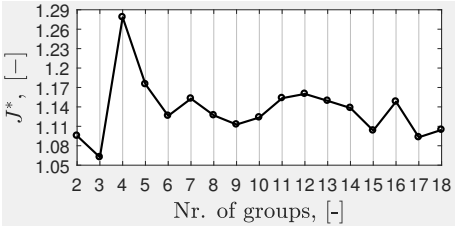
The generator slow coherency methodology in Section 5.3.2 has been evaluated on the two test systems whose electromechanical models are available in PST [83]. At first, the evaluation approach is illustrated in detail on the NPCC 48 machine test system, and then some parts of it are repeated on the IEEE 145 bus test system to show the importance of considering the row magnitudes in **Z**.

NPCC 48 MACHINE TEST SYSTEM

The NPCC 48 machine test system [186] contains 140 buses, 48 of which are buses with synchronous machines. It represents the parts of the electric power grid in the Northeastern U.S. and Southeast Canada. A geographical diagram of this system (a copy from [34]) is shown in Figure 5.11, with the dots representing the locations of the synchronous



(a) Eigenvectors of original $\mathbf{M}^{-1}\mathbf{K}$



(b) Eigenvectors of $\mathbf{M}^{-1}\mathbf{K}^{\mathbf{B}}$

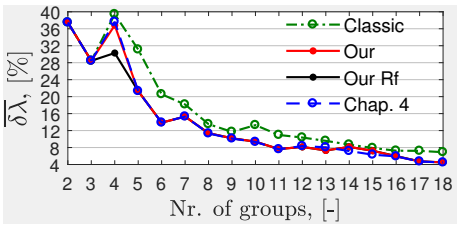
Figure 5.12: Variation of alignment cost (4.5) for the mass-normalized electromechanical eigenvectors of the NPCC 48 machine test system

machines. The NPCC 48 machine test system has been frequently used in generator coherency studies [34, 35, 121, 182, 183]. Its electromechanical model data is available in PST [83], and the matrix $\mathbf{M}^{-1}\mathbf{K}$ of this system can be found in [34, Appendix A]. The smaller 39 bus New England test system (see Figure 5.1) is partly included into the PST model of the NPCC 48 machine system, in which it corresponds to machines 1–8.

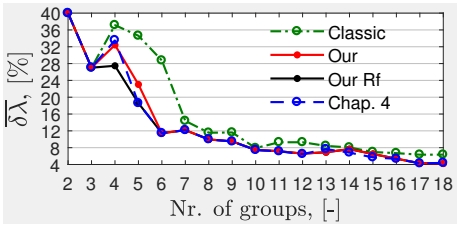
First, the alignment cost minimization method described in Section 5.3.2 is applied to the first $2,\ldots,18$ mass-normalized electromechanical eigenvectors of the NPCC 48 machine system, with the result being shown in Figure 5.12. In Figure 5.12, the minimal alignment cost is shown both for the eigenvectors of the original matrix $\mathbf{M}^{-1}\mathbf{K}$ in which $\mathbf{K}$ is non-symmetric and the eigenvectors of $\mathbf{M}^{-1}\mathbf{K}^{\mathbf{B}}$ in which $\mathbf{K}^{\mathbf{B}}$ is symmetric because its construction only considers susceptances in the Kron-reduced admittance matrix. The inclusion of results with $\mathbf{K}^{\mathbf{B}}$ aims to show the ideal outcome of the spectral clustering based coherency algorithm, which ideally requires a symmetric similarity matrix. Noteworthy, one of the local minima of the alignment cost curve in Figure 5.12 corresponds to grouping the machines into 9 areas – the choice that was previously advocated in [34, 35, 182]. Other minima correspond to 3, 6 and 15–17 areas.

Considering too many eigenvectors may not be meaningful, as faster eigenvectors of $\mathbf{M}^{-1}\mathbf{K}$ increasingly characterize local electromechanical modes, while the applications of coherency analysis (e.g., dynamic model reduction or ICI) typically prioritize slow inter-area oscillation modes. Coincidentally, choosing too few electromechanical eigenvectors may not be sufficient to convey enough details about the network structure. For example, the first and second largest eigenvalue gaps (2.20) of the IEEE 68 bus test system suggest respectively a two and three area grouping, but instead the number of generator groups is generally chosen to be 5 (see the nominal areas in Figure 5.4), which corresponds to a *local minimum* of the eigengap curve (e.g., see [34, Chapter 8]). In general, selecting a good number of machine groups may depend on the application and require some specific knowledge of the analyzed power system [187]. As previously mentioned, only the machine grouping using the slowest electromechanical eigenvectors (i.e., slow coherency grouping) is considered, with other grouping frameworks (e.g, the synchrony-based grouping [188]) being omitted to adhere to the initially defined scope.

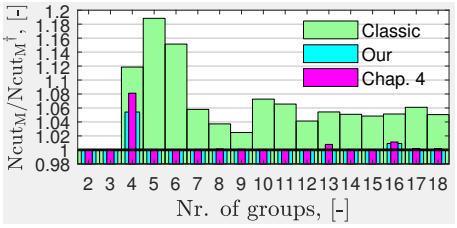
To demonstrate the effectiveness of the algorithm in Section 5.3.2, its results are compared with those of the classic slow coherency grouping algorithm [33, 34]. These algo-



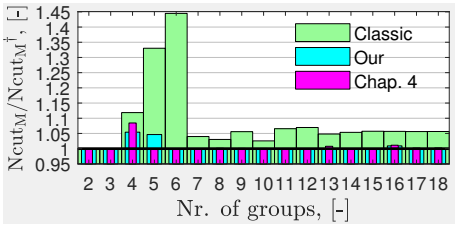
(a) $\mathbf{M}^{-1}\mathbf{K}$ as electromechanical model



(b) $\mathbf{M}^{-1}\mathbf{K}^{\mathbf{B}}$ as electromechanical model



(c) $\mathbf{M}^{-1}\mathbf{K}$ as electromechanical model



(d) $\mathbf{M}^{-1}\mathbf{K}^{\mathbf{B}}$ as electromechanical model

Figure 5.13: Mean eigenvalue errors of inertial aggregate models based on groupings of the NPCC 48 machine system produced by four slow coherency grouping algorithms and associated generic normalized cuts

rithms are directly comparable, as they both can return the specified number of groups k by using the k slowest electromechanical eigenvectors. In addition, the results by the algorithm in Section 4.4 are included to illustrate the benefits of the approach in Section 5.3.2 to slow coherency identification. To achieve better results on problem (5.11), the algorithm in Section 4.4 accepts electromechanical eigenvectors as input, minimizes generic expansions (5.12) and uses the eigenvector threshold value of $\beta = \sqrt{2}/2$. The mean percentage eigenvalue errors (5.17) resulting from applying the tested grouping algorithms to the NPCC 48 machine system for $k = 2, \ldots, 18$ are shown in Figure 5.13.

The upper plots in Figure 5.13 show that the algorithms based on alignment of spectral embedding with the canonical axes perform equally or better than the classic grouping algorithm (the dash-dot green curve) in all test cases. For the current case study, the algorithm from Section 4.4 (the dashed blue curve) is able to produce nearly the same results as the grouping algorithm consisting of Algorithm 5.1 and the greedy assignment of remaining machines (the solid red curve). The three algorithms mentioned so far have a peak in their eigenvalue error curves at $k = 4$, which also corresponds to the maximum of alignment cost in Figure 5.12. This high alignment cost value implies that the four slowest electromechanical eigenvectors provide a limited amount of information to minimize the $\mathrm{Ncut_M}$ criterion. Therefore, the Ncut refinement algorithm becomes beneficial to find lower $\mathrm{Ncut_M}$ values using the initial information contained in the eigenvectors. For $k = 4$, the complete Ncut minimization approach of Section 5.3.2 (the solid black curve) is able to noticeably improve the machine grouping in terms of metric (5.12).

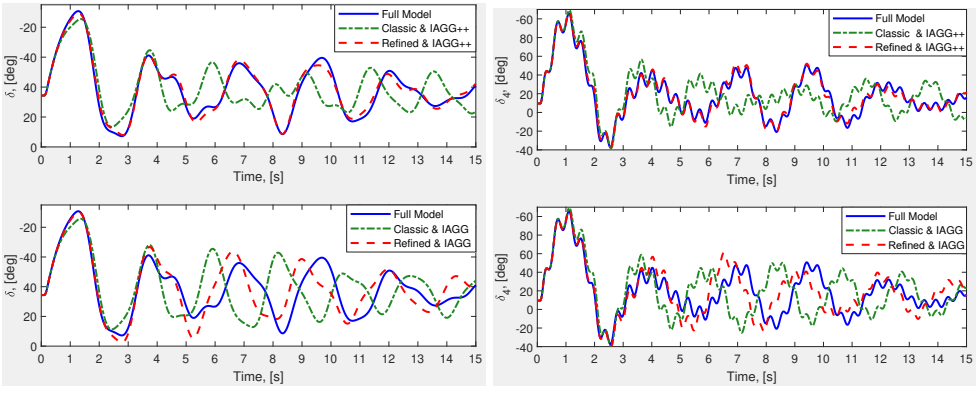
For $k = 9$, the classic grouping algorithm and the algorithm in Section 5.3.2 show nearly the same results, except machine 42 being grouped by the algorithm in Section

Full Model	Classic algorithm [33, 34]		Algorithm in Section 5.3.2	
Frequency, [Hz]	Frequency, [Hz]	$\delta\lambda$, [%]	Frequency, [Hz]	$\delta\lambda$, [%]
0.2697	0.2994	10.98	0.2851	5.703
0.3813	0.4619	21.15	0.4344	13.93
0.4876	0.5706	17.03	0.5360	9.925
0.5329	0.7174	34.60	0.6781	27.25
0.7069	0.8403	19.03	0.7945	12.55

Table 5.3: Approximation of slow modes by inertial aggregate models for the two six-area groupings

5.3.2 together with machines {32, 37, 38, 40, 41} instead of machine 39 (machine 39 thus becomes a single-machine group). The mentioned nine area grouping of the NPCC 48 machine test system can be found in multiple references (e.g., in [34, 35, 182, 183]). However, for the grouping based on the six slowest electromechanical modes (i.e., another local minimum in Figure 5.12), the results obtained from the two discussed algorithms differ substantially, which is reflected in the higher eigenvalue error returned by the classic grouping algorithm. The frequencies of the 6 slowest modes (excluding the 0 Hz mode) are given in Table 5.3 for the full electromechanical model $\mathbf{M}^{-1}\mathbf{K}$ and the corresponding inertial aggregate electromechanical models obtained from the both six area groupings. The actual areas returned by the both algorithms are shown in Figure 5.11, from which it can be seen that the proposed algorithm has returned more compact areas in terms of machines' geographic coordinates.

To conclude this case study, the obtained two six area groupings are evaluated by comparing the disturbance responses of the two corresponding nonlinear reduced models obtained with the inertial aggregation algorithm [35, 183] and its improved version in Algorithm 5.2. The study setup is assumed to be the same as in [35, 183]: for the both groupings the study area is chosen to be the area containing machines 1–8, and the ma-



(a) Time response of machine 1

(b) Time response of machine 4

Figure 5.14: Time response of machines 1 and 4 for the Medway disturbance

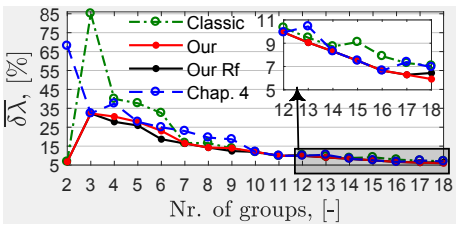
chines in the remaining five areas are replaced by the five inertial aggregate generators, while the study area is retained in full detail. The simulated event is the Medway disturbance consisting in a 6-cycle short circuit at the Medway bus (bus 7 in the `datanp48.m` file of PST), which is cleared by removing the line from Medway to the Sherman Road bus (bus 6 in the `datanp48.m` file of PST). For this event, the time responses of machines 1 and 4 are shown in Figure 5.14.

In Figure 5.14, the upper plots in each subfigure show the outcomes of Algorithm 5.2, while the lower plot shows the outcomes of the original inertial aggregation method [35, 183]; the red graphs are based on the areas obtained with our grouping algorithm, while the green graphs are based on the areas returned by the classic grouping algorithm, and the blue graphs show the response of the unreduced system. As it can be seen, the six area grouping produced by the proposed grouping algorithm results in a noticeably smaller difference in the simulated rotor angle curves between the original and reduced models. If Algorithm 5.2 is used for generator aggregation, these curves practically coincide. This result comes as no surprise when looking at the underlying machine groups for the two reduced models in Figure 5.11, as the proposed grouping algorithm has returned more geographically compact electromechanical areas. Besides the Medway disturbance, several other disturbances have been simulated in the chosen study areas, and the conclusions drawn from Figure 5.14 remained valid for all of them.

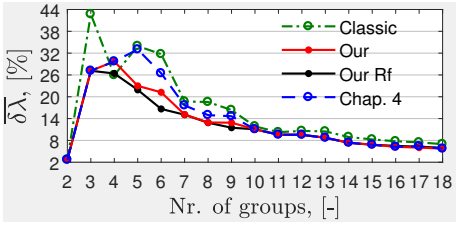
IEEE 145 BUS TEST SYSTEM

The IEEE 145 bus 50 machine test system is another test power system, the electromechanical model of which is available in PST. Unfortunately, no single line diagram or geographical diagram of this network could be found in open sources to provide a more complete description. Using this system as an example, the impact of the differences between the approaches in Section 5.3.2 and 4.4 is demonstrated more explicitly than in the previous case study on the NPCC 48 machines test system.

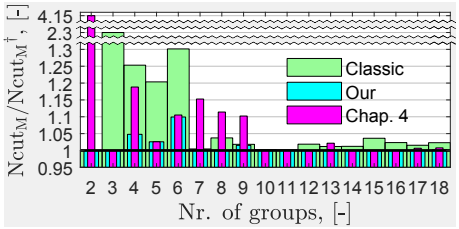
The comparison of eigenvalue approximation accuracy by different machine grouping methods is repeated for the IEEE 145 bus test system, with the corresponding results being presented in Figure 5.15a. By looking at Figure 5.15, a significant eigenvalue approximation error due to using the partitioning algorithm from Section 4.4 can be easily noticed for $k = 2$. However, the same error is not present for the analogous test case in Figure 5.15b. The machine grouping for $k = 2$ resulting in the low eigenvalue errors of respectively 6% and 2% is the same for the both models $\mathbf{M^{-1}K}$ and $\mathbf{M^{-1}K^B}$, and it consists in classifying machine 43 at bus 137 into a separate group. For the model $\mathbf{M^{-1}K^B}$, machine 43 is very well separated in spectral embedding from the other machines, which results in the correct grouping by the algorithm in Section 4.4 despite normalizing each row of $\mathbf{Z}$ to unity length. However, when the slowest two eigenvectors of $\mathbf{M^{-1}K}$ are used instead of those of $\mathbf{M^{-1}K^B}$, the resulting spectral embedding exhibits a more disperse structure. Machine 43 still remains separated from the rest due to the high magnitude of the row of $\mathbf{Z}$ corresponding to it. However, if the rows of $\mathbf{Z}$ are normalized to length one as the algorithm in Section 4.4 requires, considering only the *angular separation* of machine 43 from the rest becomes not enough to identify it as a separate group. As the result, Figure 5.15a shows a substantial eigenvalue error by the algorithm in Section 4.4 for $k = 2$, and Figure 5.15c reveals that this error is linked to a very high relative $\mathrm{Ncut_M}$ value of 4.15. Although the algorithm in Section 4.4 still performs relatively well in many



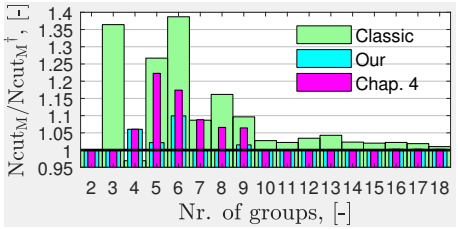
(a) $\mathbf{M}^{-1}\mathbf{K}$ as electromechanical model



(b) $\mathbf{M}^{-1}\mathbf{K}^{\mathbf{B}}$ as electromechanical model



(c) $\mathbf{M}^{-1}\mathbf{K}$ as electromechanical model



(d) $\mathbf{M}^{-1}\mathbf{K}^{\mathbf{B}}$ as electromechanical model

Figure 5.15: Mean eigenvalue errors of inertial aggregate models based on groupings of the IEEE 145 bus test system produced by four algorithms and associated generic normalized cuts.

test cases based on the IEEE 145 bus test system, its results become noticeably worse compared to the coherency grouping algorithm proposed in this chapter. The reason lies in the ability of the method in Section 5.3.2 to use both the angular and radial separation of the rows of **Z**.

5.4. CONCLUSIONS

This chapter considered the two prominent power system applications that involve decomposition of a power system into control zones or areas. The application of the clustering framework in Chapter 4 (including its modifications) was shown to result in noticeable accuracy improvements in identifying both the VCZs for SVC and the slow coherent machine groups, thus contributing to research question I (see Section 1.2.3).

Regarding the task of VCZ definition for SVC, the VCS zoning method previously proposed in [52, 173] was adapted to be used together with the framework in Chapter 4, thus avoiding the drawbacks inherent in VCZ methods based on hierarchical clustering [46, 52, 173]. The test results on the IEEE 39 bus test system have demonstrated that the proposed spectral VCS method is able to better reveal the relevant *generator voltage domains*, which allowed to significantly facilitate the SVC pilot bus selection ensuring a robust voltage profile across the whole power network. Although this chapter only includes the VCZ results based on the VCS concept, some other SVC zoning methods from literature can be seamlessly adapted for their use with the framework in Chapter 4 as well. For example, the framework in Chapter 4 can be directly used with voltage sensitivity matrices from [46] and [47, 51, 93]. Various voltage sensitivity matrices may entail

somewhat different requirements to pilot buses. For example, the VCS sensitivity matrices (5.4) and (5.7) empathize the pilot bus controllability by the VCS control generators, while the voltage sensitivity matrix used for SVC zoning in [47, 51, 93] rather empathizes the observability of voltage disturbances by the pilot buses.

The second part of this chapter dealed with the slow coherency problem that is relevant for power system dynamic model reduction, DSA, and the design of certain WAMPAC solutions (e.g., ICI) [35, 55]. The classic slow coherency problem was shown to be closely related to a generic normalized cut minimization problem [189] through spectral relaxation. Considering this, the previously introduced metrics of cluster expansion (2.23) and normalized cut (2.7) receive an additional meaning for power systems. The connection between slow coherency and normalized cuts prompted the discovery of the role of mass-normalized electromechanical eigenvectors in revealing the orthogonal structure inherent in the slow coherency grouping problem. To solve the problem in (5.11) more effectively, a new problem-specific grouping algorithm was developed in Section 5.3.2. This algorithm was shown to consistently produce more accurate machine groupings than other comparable methods including the classic algorithm in [33, 34] and the initial algorithm in Section 4.4.

Although the Ncut_M criterion in (5.11) was shown to be an efficient indicator of the slow coherency grouping accuracy, the demonstrated relationship between the two is neither linear nor monotonic. For some test cases, a moderate increase in eigenvalue approximation errors (5.17) was observed after achieving a slight decrease of the Ncut_M value after applying the Ncut_M refinement algorithm. However, achieving the lowest possible Ncut_M values still appears to be a powerful heuristic to identify power system areas based on slow coherency.

6

CONSTRAINED GRAPH PARTITIONING AND ICI

6.1. INTRODUCTION

This chapter concerns Intentional Controlled Islanding (ICI), which is the second of the two WAMPAC applications detailed in Section 2.2[1]. In what follows, the static voltage stability requirement will be ignored among the eight ICI objectives listed in Section 2.2.2. Neglecting this constraint allows to simplify the ICI power flow constraints from transcendental nonlinear AC power flow equations to linear DC power flow equations. This leads to conversion of the original mixed-integer nonlinear programming (MINLP) problem into a MILP problem that can be solved more efficiently due to huge advances in commercial MILP solvers over the past few decades. Neglecting the transcendental nature of AC power flow equations implies the assumption of sufficient reactive power reserves in each island to maintain a high voltage profile after system splitting (as shown in [190], even this assumption may not always be enough). However, the main goal of using simplified ICI representation such as the direct current (DC) OPF MILP model or the active power flow graph partitioning model is in improving the performance of resolving the ICI constraints 1–6 and 8 of Section 2.2.2 that remain important building blocks of more complex and accurate ICI models.

The major contribution of this chapter is a polynomial-time heuristic algorithm for the NP-hard *packing Steiner trees* problem [77], which lies at the heart of ensuring the generator coherency and island connectivity constraints. The proposed heuristic algorithm substantially differs from the previously published approaches [57, 58, 59, 149], as it attempts to connect the coherent generators one-by-one in a parallel and bottom-up fashion, starting from generators that are closest to each other. The distances between generators can be computed from constrained spectral embedding generated based on the flexible constrained spectral clustering [191] (FCSC) method [191, 192] that was pre-

[1] The material of this chapter is partly based on [J2].

viously used in the power system context in [60]. Compared to [60], the proposed algorithm explicitly handles the connectivity of the resulting partitions.

Interconnecting coherent generators in each island and minimizing the power flow disruption between islands can be considered as an efficient and effective ICI strategy, and multiple heuristic approaches have been proposed to solve this problem in polynomial time [43, 57, 58, 59, 149]. Similar problem formulations involving assignments of certain *terminal nodes* to specific graph partitions also appear in other disciplines (e.g., machine learning [126, 191] and VLSI design [77, 91]). However, it is understandable that network splitting solutions that only minimize power flow disruption, including the polynomial-time algorithm proposed in this chapter, may not satisfy all the static ICI constraints listed in Section 2.2.2. In other words, opening lightly loaded lines for system splitting promotes small load-generation imbalances and small equipment overloads, but it cannot guarantee them. Nevertheless, quickly finding a solution with a low MW cut that satisfies the island connectivity and coherency grouping constraints may be a good starting point for searching the global optimal solution of a more precise NP-hard MILP (or MINLP) formulation.

The following section will describe the computation of constrained spectral embeddings using the FCSC algorithm [191]. Graph distances induced by such embeddings can serve as an input to the polynomial-time heuristic for the packing Steiner trees problem that is proposed in Section 6.3.1. Next, Section 6.4 shows the application of the algorithm proposed in Section 6.3.1 as a method to speed up the MILP solvers by providing them with good initial solutions. Finally, Section 6.5 summarizes this chapter.

6.2. FCSC-BASED CONSTRAINED SPECTRAL EMBEDDING

Unlike the variants of spectral clustering considered in Chapters 3–5, flexible constrained spectral clustering [191] (FCSC) extends spectral clustering to incorporate node grouping constraints into its formulation. Other methods with a similar background include [69, 126, 193, 194], to name a few. The main reasons to choose FCSC over other alternatives are the implicit generation of a geometric graph embedding and the intuitive extension to k-way clustering. The other advantages of FCSC include the simplicity of implementation and the close relationship to the previously considered spectral clustering formulations.

The overall FCSC method has been published in [191, 192]. Therefore, only a brief summary of the algorithm is included here. The pairwise *must-link* and *cannot-link* constraints introduced in Section 3.4.1 can be encoded into the *constraint matrix* $\mathbf{Q} \in \{-1, 0, 1\}^{n \times n}$ defined element-wise as follows:

$$[\widetilde{Q}_{ij}] = [\widetilde{Q}_{ji}] = \begin{cases} +1, & \text{if ML}(v_i, v_j) \\ -1, & \text{if CL}(v_i, v_j) \\ 0, & \text{if no ML or CL constraint between } v_i \text{ and } v_j \end{cases} \tag{6.1}$$

where $\text{ML}(i, j)$ and $\text{CL}(i, j)$ indicate the must-link and cannot-link constraints between nodes v_i and v_j.

Assuming a two cluster partitioning, it can be described by a single cluster indicator vector $\widetilde{\boldsymbol{X}} \in \{-1, 1\}^n$, where $\widetilde{x}_i = 1$ if node v_i belongs to cluster $\mathcal{C}_1$ and $\widetilde{x}_j = -1$ if node v_j

belongs to cluster $\mathcal{C}_2$. Then the satisfaction of constraints encoded in the matrix $\mathbf{Q}$ can be expressed as

$$\widetilde{\boldsymbol{X}}^T\mathbf{Q}\widetilde{\boldsymbol{X}} = \sum_{i=1}^{n}\sum_{j=1}^{n}\widetilde{x}_i\widetilde{x}_j Q_{ij} \tag{6.2}$$

where the measure in (6.2) increases by one for each satisfied constraint and decreases by one for each violated constraint in $\mathbf{Q} \in \{-1,0,1\}^{n\times n}$.

The constraint matrix $\mathbf{Q}$ in (6.1) can also be relaxed as $\mathbf{Q} \in \mathbb{R}^{n\times n}$ to express the *degrees of belief* in individual pairwise constraints (i.e., $-1 \le Q_{ij} \le 1$). For simplicity, only constraint matrices $\mathbf{Q} \in \{-1,0,1\}^{n\times n}$ are used in this chapter. Furthermore, the *normalized constraint matrix* $\mathbf{Q_n}$ and the scaled indicator vector $\widetilde{\boldsymbol{U}}$ are introduced to closely follow the derivations in [191, 192] that are largely omitted here:

$$\mathbf{Q_n} = \mathbf{D}^{-\frac{1}{2}}\mathbf{Q}\mathbf{D}^{-\frac{1}{2}} \tag{6.3}$$

$$\widetilde{\boldsymbol{U}} = \mathbf{D}^{\frac{1}{2}}\widetilde{\boldsymbol{X}} \tag{6.4}$$

where $\mathbf{D}$ is the degree matrix (2.2).

With the above definitions, the FCSC optimization problem for two clusters is stated in [191, 192] as follows:

$$\text{minimize} \quad \boldsymbol{U}^T\mathbf{L_n}\boldsymbol{U} \tag{6.5a}$$

$$\text{subject to:} \quad \boldsymbol{U} \in \mathbb{R}^n,\ \boldsymbol{U}^T\mathbf{Q_n}\mathbf{U} \ge \alpha,\ \boldsymbol{U}^T\boldsymbol{U} = \text{vol}(\mathscr{G}),\ \boldsymbol{U} \ne \mathbf{D}^{\frac{1}{2}}\mathbf{1}_{n\times 1} \tag{6.5b}$$

where $\boldsymbol{U}$ is the relaxed cluster indicator vector $\widetilde{\boldsymbol{U}}$ in (6.4), $\mathbf{L_n}$ is the symmetric normalized Laplacian (2.6c) of the similarity graph $\mathscr{G}$ describing the power network structure (e.g., a branch admittance graph or an active power flow graph described in Section 3.2.1), $\text{vol}(\mathscr{G})$ is the volume of the graph $\mathscr{G}$ (2.4), and α is a user-specified lower threshold on how well the constraints in $\mathbf{Q}$ are satisfied (note that $\widetilde{\boldsymbol{X}}^T\mathbf{Q}\widetilde{\boldsymbol{X}} = \widetilde{\boldsymbol{U}}^T\mathbf{Q_n}\widetilde{\boldsymbol{U}}$).

The problem in (6.5) is tackled in [191, 192] by applying the Karush-Kuhn-Tucker (KKT) conditions. After several mathematical steps, it is determined in [191, 192] that solutions of (6.5) can be found as the generalized eigenvectors of (6.6a) that satisfy the conditions (6.6b), (6.6c), and (6.6d):

$$\mathbf{L_n}\boldsymbol{U} = \lambda\left(\mathbf{Q_n} - \frac{\alpha'}{\text{vol}(\mathscr{G})}\right)\boldsymbol{U} \tag{6.6a}$$

$$\lambda > 0 \tag{6.6b}$$

$$\boldsymbol{U}^T\mathbf{Q_n}\boldsymbol{U} \ge \alpha' \tag{6.6c}$$

$$\boldsymbol{U}^T\boldsymbol{U} = \text{vol}(\mathscr{G}) \tag{6.6d}$$

where α' is the modified lower threshold on how well the constraints in $\mathbf{Q}$ are satisfied. It is proven in [191, 192] that $\alpha' < \alpha$ always holds, which means that the desired value of $\boldsymbol{U}^T\mathbf{Q_n}\boldsymbol{U} = \alpha > \alpha'$ (i.e., the algorithm's bias towards satisfying the constraints in $\mathbf{Q}$) can be exceeded by specifying α' in (6.6a).

Higher values of α' in (6.6) cause a higher bias towards satisfying the constraints in $\mathbf{Q}$. However, setting α' too high may result in no feasible solutions satisfying the condition

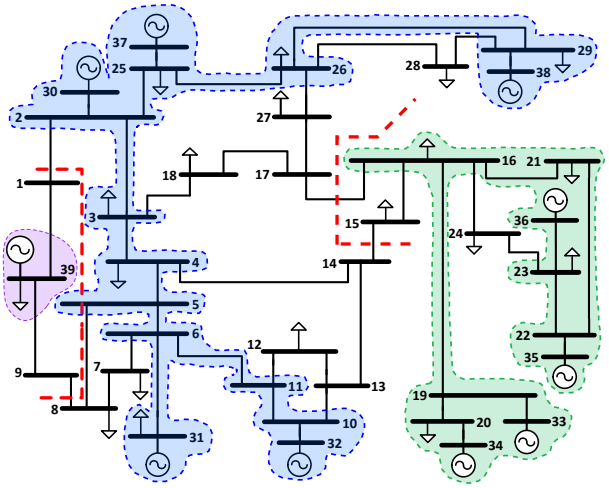


Figure 6.1: IEEE 39 bus test system. Group trees obtained with Algorithm 6.1 are shown in colors. The red dashed lines denote the final cutset.

(6.6b) (i.e., all generalized eigenvalues of (6.6a) may become non-positive). According to [191, 192], to ensure at least one feasible generalized eigenvector $\boldsymbol{U}$ of (6.6a), the following condition on α' should hold:

$$\alpha' < \nu_{max}\,\mathrm{vol}(\mathscr{G}) \tag{6.7}$$

where ν_{max} is the largest eigenvalue of $\mathbf{Q_n}$.

In [192], the relaxation (6.6) is explicitly extended to the case when $k > 2$ by requiring at least $k-1$ eigenpairs of the generalized eigenproblem in (6.6a) to satisfy (6.6b), (6.6c), and (6.6d). To ensure the existence of $k-1$ feasible eigenpairs, the condition (6.7) should be modified as follows:

$$\alpha' < \nu_{k-1}\,\mathrm{vol}(\mathscr{G}) \tag{6.8}$$

where ν_{k-1} is the $k-1$-th largest eigenvalue of $\mathbf{Q_n}$.

No explicit derivation has been provided in [192] to justify the extension of (6.6) and (6.7) for $k = 2$ to (6.6) and (6.8) for $k > 2$. In particular, the indicator vector representation $\widetilde{\boldsymbol{X}} \in \{-1, 1\}^n$ that underpins (6.5) and its relaxation (6.6) is straightforward only for $k = 2$ (see also [74, 123, 127]). However, the following sections will demonstrate that extending the initial FCSC algorithm in [191] according to (6.8) can produce spectral k-embeddings that amplify the input constraints in $\mathbf{Q}$. This outcome is sufficient to formulate constrained graph partitioning algorithms that can use constrained graph embeddings (e.g., the algorithm in Section 6.3.1). The derivation of geometric graph embeddings that optimally represent the graph connectivity structure *and* the node grouping constraints is outside the scope of this chapter.

To illustrate the generation of constrained graph embeddings by the FCSC algorithm described in this section, consider the following example based on the IEEE 39 bus test power system [180]. The single-line diagram of the IEEE 39 bus test power system corresponding to this case study is shown in Figure 6.1. The contents of Figure 6.1 imply

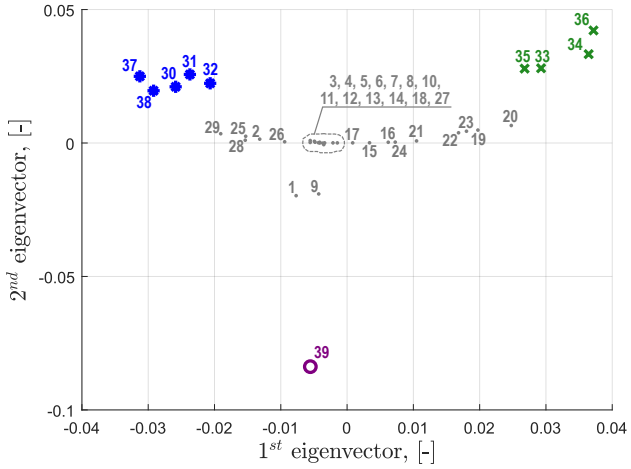


Figure 6.2: Constrained graph embedding of the IEEE 39 bus system computed with FCSC for the generator groups in Figure 6.1 and branch admittances as graph edge weights

the presence of three groups of terminal nodes: $\mathcal{T}_1 = \{39\}$, $\mathcal{T}_2 = \{30,31,32,37,38\}$, and $\mathcal{T}_3 = \{33,34,35,36\}$. This experimental grouping was obtained by clustering the rows of the eigenvector matrix composed of the second and third slowest electromechanical eigenvectors with AHC using average linkage as explained in [J2]. The sets of terminal nodes $\mathcal{T}_1$, $\mathcal{T}_2$, and $\mathcal{T}_3$ define the constraint matrix $\mathbf{Q} \in \{-1,0,1\}^{39\times 39}$, in which ones correspond to bus pairs from the same group $\mathcal{T}$ and minus ones correspond to bus pairs not in the same group $\mathcal{T}$, $\mathcal{T} \in \{\mathcal{T}_1,\mathcal{T}_2,\mathcal{T}_3\}$. The similarity graph, for which the normalized Laplacian matrix $\mathbf{L_n}$ and the degree matrix $\mathbf{D}$ are computed, is assumed to be the branch admittance graph (see Figure 4.3). The FCSC parameter α' is set to $0.5(v_2+v_3)\mathrm{vol}(\mathscr{G})$. For this input data, the constrained graph embedding obtained with FCSC is presented in Figure 6.2 after normalizing the generalized eigenvectors $\boldsymbol{U}$ computed from (6.6), (6.8) to length one and applying the inverse of transformation (6.4) to the both eigenvectors.

6.3. CONSTRAINED GRAPH PARTITIONING

This section presents an algorithm for partitioning of weighted undirected graphs subject to node grouping constraints. Given k disjoint sets of terminal nodes $\mathcal{T}_1,\ldots,\mathcal{T}_k$ that should be placed each into a separate graph partition, one way to satisfy the node grouping constraints while ensuring island connectivity is to construct k disjoint trees $\mathscr{T}_1,\ldots,\mathscr{T}_k$ in the graph $\mathscr{G}$, each spanning its own set of terminal nodes. Such tree subgraphs each spanning only a subset of all graph nodes are known as *Steiner trees*. Thus, the algorithm's primary goal is to construct k disjoint Steiner trees for the k disjoint sets of terminal nodes given as an input, which constitutes the packing Steiner trees problem described in [77].

After the Steiner trees have been constructed, each of them can be merged into a single node as described in Section 3.4.1, which ensures the connectedness of the node sets $\mathcal{T}_1,\ldots,\mathcal{T}_k$. The remaining task is to produce a contiguous partitioning of the reduced

graph that separates the k merged nodes from each other while minimizing the total graph cut (2.22), which was detailed e.g. in [C1] (in general, this problem is known as k-way cut in literature). Finally, the partitioning of the reduced graph can often be noticeably improved by using various post-processing algorithms (e.g., the label propagation based cut refinement algorithm in Section 3.3.2).

6.3.1. SEQUENTIAL TREE GROWING ALGORITHM

The proposed tree construction algorithm shares some conceptual similarity with the AHC using single linkage [71], which has also been used in Section 3.4.2. It starts by placing each terminal node as a separate *subcluster* belonging to its own prospective partition, and at the end all initial subclusters should be connected through a series of subcluster merges to form k disjoint Steiner trees. At any stage of the subcluster merging process, the union of subclusters of the same prospective partition is further referred to as *group cluster*, and it becomes a group Steiner tree when all underlying subclusters become connected. The two subclusters are merged by computing the shortest path connecting them and including the intermediate path nodes as well as the nodes of the two subclusters into the newly merged subcluster. Shortest paths can be computed on various *distance graphs* whose interconnection structure coincides with the power network topology (e.g., graphs based on FCSC embedding or branch impedance graphs).

start path node	end path node	path total length	group cluster id	start subcluster id	end subcluster id	{path nodes}

Figure 6.3: Storing format for paths between subclusters

In the beginning, the shortest paths between all subclusters inside each group cluster are computed, and each path is stored in the format shown in Figure 6.3. In Figure 6.3, *path total length* is the sum of edge weights in the distance graph along the stored path from the start path node and to the end path node; the *group cluster id* entry denotes the index of the group cluster to which the start and end path nodes belong; the *start subcluster id* and *end subcluster id* entries denote the indices of subclusters of the current group cluster to which the start and end path nodes belong. Thus, every computed path uses at least the first six entries of the path format in Figure 6.3. If the start path node and the end path node are not directly connected, the intermediate nodes between them are stored in the entries following the end subcluster id entry. After a path is computed and stored in the format described above, it is put into a priority queue [159] with the path total length being the priority index (a lower distance means a higher priority).

As it is not admissible for a path between the nodes of one group cluster to go through the nodes of another group cluster, a binary edge status matrix $\widetilde{\mathbf{E}}^{m\times k}$ is created to keep track of the edges that are adjacent to the nodes currently contained in each group cluster. Before starting to compute paths for a group cluster, this matrix is used to disable the edges adjacent to the nodes of the other group clusters, thus precluding the computed paths to go through them. As power networks are by their nature sparse graphs with the number of branches not being significantly higher than the number of buses, storing an $m \times k$ binary matrix is computationally feasible.

Algorithm 6.1 Sequential Tree Growing

Input: Distance graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$, sets of terminal nodes $(\mathcal{T}_1, \ldots, \mathcal{T}_k)$

1: $\mathbf{W} \leftarrow$ Adjacency matrix of $\mathcal{G}$ // $\mathbf{W}$ includes distances as edge weights
2: PQ1 $\leftarrow$ Empty priority queue // Stores paths
3: PQ2 $\leftarrow$ Empty priority queue // Stores paths to be reconsidered later
4: $\widetilde{\mathbf{E}} \leftarrow$ Set $\widetilde{E}_{ij}$ to 1 for edges adjacent to the nodes in $\bigcup_{l=1}^{k} \mathcal{T}_l \setminus \mathcal{T}_j,\ j = 1, \ldots, k$
5: $\mathbf{N} \leftarrow \mathbf{0}_{n \times k}$
6: Initialize k union-find structures to contain the nodes in each of the sets $\mathcal{T}_1, \ldots, \mathcal{T}_k$ as initial subclusters. // union-find structure is detailed in Algorithm 3.6
7: Compute the initial paths between the subclusters of each group cluster. Use $\widetilde{\mathbf{E}}$ to constrain the paths in each group cluster to bypass the nodes in other group clusters. Put the resulting paths into PQ1. Update $\mathbf{N}$ after computing each path.
8: **while** any of group clusters is disconnected **do**
9: **if** PQ1 is empty **then**
10: **if** PQ2 remains the same or PQ2 is empty **then**
11: Compute new paths by isolating the nodes with two or more nonzero entries in $\mathbf{N}$ (in addition to ordinary path constraints using $\widetilde{\mathbf{E}}$) and add the new paths to PQ1. If no new feasible paths could be found or no nodes with two or more nonzero entries in $\mathbf{N}$ exist, break.
12: **else**
13: Copy PQ2 into PQ1. Re-initialize PQ2 to an empty priority queue.
14: **end if**
15: **end if**
16: Extract the smallest distance path out of PQ1.
17: **if** *start subcluster id* and *end subcluster id* are the same **then**
18: Decrement the entries of $\mathbf{N}$ corresponding to the path nodes and group cluster of the current path by one.
19: **continue**
20: **end if**
21: $path \leftarrow$ current path's nodes including *start path node* and *end path node*.
22: **if** $\mathbf{N}[path; 1, \ldots, k]$ has only one nonzero column **then**
23: Merge the two subclusters connected by the current path.
24: Compute new paths from the path nodes of the current path to the nodes of all subclusters of the current group cluster except the nodes of the newly merged subcluster. $\widetilde{\mathbf{E}}$ is used to constrain the paths as in line 7. Update $\mathbf{N}$ after computing each new path. Put the new paths into PQ1.
25: Update $\widetilde{\mathbf{E}}$ by indicating the edges adjacent to the path nodes of the current path.
26: **else**
27: Put the extracted path into PQ2.
28: **end if**
29: **end while**
30: If some group clusters still remain disconnected, greedily merge the minor subclusters to the major ones, starting from the minor subclusters having the largest number of terminal nodes.

Output: Steiner trees $\mathcal{T}_1, \ldots, \mathcal{T}_k$.

Additionally, an integer matrix $\mathbf{N}_{n\times k}$ is created to store the *number of paths* that pass through each of the n graph nodes for each of the k group clusters. After a path between some subclusters of group cluster j is computed, the entries of $\mathbf{N}$ in column j and the rows corresponding to the path nodes are incremented by one, and the path is added to `PQ1`. If a path extracted from `PQ1` connects the two nodes that are already in the same subcluster (i.e., a connection between the nodes has already been made through another path), the entries of $\mathbf{N}$ in the rows corresponding to the path's nodes and the column corresponding to path's group cluster are decremented by one, which means that the path should be excluded from consideration for being superfluous. If all rows of $\mathbf{N}$ corresponding to nodes of some path share the same single non-zero column, such path does not contradict any computed paths connecting other group clusters, and it is acceptable to merge the two subclusters. However, if some path node has two or more non-zero entries in the matrix $\mathbf{N}$, such node could be possibly required for the connectivity of other group clusters, so the involved path is put into the second priority queue `PQ2` that serves to collect the paths to be reconsidered later.

The merging of subclusters and the identification of subcluster of each node is realised with the union-find data structure (see [159] and Algorithm 3.6). For each group cluster, a separate union-find structure is created that contains the terminal nodes of the group as the initial subclusters. Given the above information, the tree construction algorithm can be summarized as Algorithm 6.1. After the k group Steiner trees have been constructed with Algorithm 6.1, they can be merged to k single nodes. These nodes can be put each into a separate connected partition by the approaches described in the beginning of this section.

6.3.2. EVALUATION OF CONSTRAINED GRAPH PARTITIONING

This section compares the proposed constrained partitioning method with the two state-of-the-art alternative methods that are able to handle terminal nodes constraints. As the proposed tree construction algorithm is mostly focused on assigning all terminal nodes requested to be in the same group to a single connected partition, the only performance metric is the number of misplaced terminal nodes (2.26).

The first benchmark algorithm hMetis [91] is a popular and highly efficient hypergraph partitioning algorithm. For each study case, it is run 25 times with the increasing values of the partition imbalance parameter (a common and important parameter for multilevel graph partitioning algorithms including hMetis), and the partitioning result with the lowest number of misplaced generators is retained. As the recursive bisection algorithm of hMetis was consistently more efficient in handling node grouping constraints, it was used in all studies. The second benchmark algorithm is based on the recent research on tight continuous relaxations of the balanced k-cut problem (TCRBGC) [126]. The online MATLAB code implementation of [126] provided by the authors was used with ratio cut as the optimization criterion, and the number of initializations to compute the final solution was set to 12 (the default value).

The benchmark power networks are taken from PST [83] and MATPOWER [81, 162]. While the power flow data is available in models from both PST and MATPOWER, only models from PST include the dynamic generator data. However, MATPOWER contains power flow data of several large-scale (over 1000 buses) power networks. These large-

scale networks are adapted for the computation of generator groups by adding the reference dynamic data from [86] according to the generator maximum power output available from MATPOWER.

In the experiments of this section, the k generator groups are obtained by clustering the rows of the eigenvector matrix composed of the k slowest electromechanical eigenvectors (excluding the first constant eigenvector) using average linkage AHC, with k being the requested number of groups[2]. While this method is less accurate than the dedicated slow coherency grouping algorithm of Section 5.3, it usually groups together generators that are close to each other in the network. Thus, the resulting node grouping constraints are consistent in the majority of cases. As no clustering algorithm is capable of always returning the correct clustering, a small fraction of generator groupings obtained by the described simple method can lead to unsatisfiable node grouping constraints. However, having generator groupings that are not straightforward (or sometimes even impossible) to fully interconnect is an advantage from the algorithm testing point of view.

The proposed constrained network partitioning algorithm has been tested on several power network models from PST and MATPOWER. In this section, the test results for the following three networks are included:

- NPCC 48 machine test system from PST, which is a 48 machine 140 buses network (see Section 5.3.5).
- *case1354pegase* test network from MATPOWER containing 1354 buses and 260 generators (see Chapters 3–4).
- *case2869pegase* test network from MATPOWER containing 2869 buses and 510 generators (see Chapters 3–4).

In the case of seven generator groups for the NPCC 48 machine test network, all three methods result in three misplaced generators (Tables 6.1–6.3) because inconsistent node grouping constraints are produced for that case. Similarly, the occasional inconsistent node grouping constraints cause the appearance of a small number of misplaced generators for the nine groups in *case2869pegase* test network. However, in all tested situations, Algorithm 6.1 consistently results in less misplaced generators (up to 18 generator groups have been tested for each network), while requiring less computational time than the two other methods.

Algorithm \ Nr. of groups	2	3	4	5	6	7	8	9	10	11	12
Sequential Tree Growing	0	0	0	0	0	3	0	0	0	0	0
hMetis	0	0	0	0	1	3	0	0	0	0	0
TCRBGC	0	0	0	0	0	3	0	0	1	0	0

Table 6.1: Number of misplaced generators for the NPCC 48 machine test system

[2] In [J2], it was decided to drop the first electromechanical eigenvector because by itself this constant vector does not help to differentiate the generators. This decision is advocated by synchrony-based generator grouping [188], while slow coherency algorithms in [33, 34] and Section 5.3 normally preserve the slowest eigenvector.

Algorithm \ Nr. of groups	2	3	4	5	6	7	8	9	10	11	12
Sequential Tree Growing	0	0	0	0	0	0	0	0	0	0	0
hMetis	2	0	0	1	0	7	0	0	0	11	17
TCRBGC	0	0	0	0	0	0	0	0	0	0	0

Table 6.2: Number of misplaced generators for the case1354pegase test system

Algorithm \ Nr. of groups	2	3	4	5	6	7	8	9	10	11	12
Sequential Tree Growing	0	0	0	0	0	0	0	2	0	0	0
hMetis	0	0	11	1	1	1	1	5	4	6	3
TCRBGC	0	0	4	2	4	5	5	38	6	5	5

Table 6.3: Number of misplaced generators for the case2869pegase test system

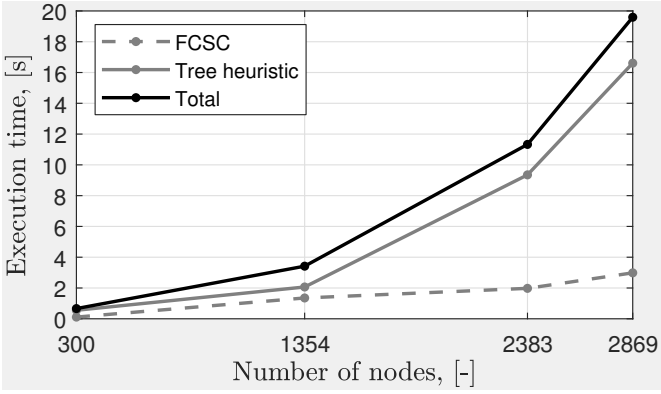


Figure 6.4: Running time of the proposed constrained graph partitioning algorithm

The higher running time of methods [91, 126] is largely caused by their multiple initializations (25 and 12, respectively) which are necessary to improve the quality of their results. The higher number of misplaced generators with methods [91, 126] is partly caused by the excessive number of connected components that can often occur as the result of partitioning. While the minor connected components could be reconnected using various heuristics (e.g., Algorithm 3.1), such an end result would be a product of the initial partitioning algorithm and a proper connectivity heuristic that should also respect the node grouping constraints.

The computational performance of the proposed constrained partitioning method is demonstrated in Figure 6.4. The results in Figure 6.4 are obtained for the MATPOWER test power networks *case300* (based on the IEEE 300 bus test power system), *case1354pegase, case2383wp* (based on the power grid of Poland), and *case2869pegase*, each containing 300, 1354, 2383, and 2869 buses respectively. The running times have been estimated with MATLAB R2017a (64-bit) on a PC with an Intel® Xeon® E5 3.70 GHz

CPU and 16 Gb of RAM on a single core. The eigenvalue calculations at the core of FCSC have been performed with an iterative sparse matrix solver (i.e., with the *eigs* function in MATLAB), by requesting to find the first eigenvalues close to a very large real number (set empirically). This approach has helped to avoid computing the full generalised eigendecomposition and significantly reduced the computation time for larger networks.

With respect to the timing results shown in Figure 6.4, it should be noted that the current MATLAB implementation of the proposed algorithm has a significant amount of possibilities for optimisation. In particular, the efficiency of implementation of the priority queue data structure may have a very large influence in the running time of the algorithm. The operations related to the currently used priority queue from the Java programming language (using the MATLAB Java API) are responsible for about 30% of the execution time of Algorithm 6.1 shown in Figure 6.4. However, this is a more than two times reduction from the values given in [J2]. This speedup could be achieved by simply changing the calling syntax of the Java priority queue object in MATLAB. Nevertheless, using more efficient data structure implementations still remains an important task.

6.4. GOOD INITIAL SOLUTIONS FOR MILP-BASED ICI

MILP allows to precisely solve many of power system constraints inherent to ICI (e.g., see the constraints in Section 2.2.2) by formulating ICI as an NP-hard discrete optimization problem. In this section, the basic familiarity with solving MILPs through branching (e.g., branch-and-bound or branch-and-cut [195]) is taken for granted. The basic model in (6.9) [112, 196, 197] is going to be used to demonstrate the impact of good initial solutions produced by Algorithms 6.1 and 3.2 on solving MILP-based ICI.

The formulation in (6.9) is based on the DC OPF assumptions that neglect the network voltage profile and reactive power, while allowing to approximate the active power relationships with a set of linear equations. These equations become linear mixed-integer if the branches in the network are allowed to be switched, as it can be seen in (6.9g). The optimization objective (6.9a) of the formulation in (6.9) consists in minimization of the total load shedding following the network splitting, where $P_{LS,i}$ is the MW load shedding amount at bus v_i. The network interconnection structure is modeled by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of graph nodes, $\mathcal{E}$ is the set of undirected graph edges, $\mathcal{T}_1, \ldots, \mathcal{T}_k$ are the sets of terminal nodes to be grouped together (e.g., k coherent generator groups), and $r_1, \ldots, r_k$ are the k *reference nodes* belonging to the corresponding terminal node sets $\mathcal{T}_1, \ldots, \mathcal{T}_k$. Constraint (6.9b) ensures that every network node should belong to one of k partitions. If node v_i belongs to block l, the binary node status variable $x_{i,l}$ is set to one and otherwise to zero. The partitioning status of the nodes belonging to a set of terminal nodes can be fixed in advance, which is expressed as (6.9c). Constraints (6.9d) introduce extra binary variables $z_{i,j,l}$ to represent the products of binary node status variables $x_{i,l}x_{j,l}$, $l = 1, \ldots, k$ by using the well-known McCormick linearization technique. Constraint (6.9e) defines the edge status variables ($y_{i,j} = 0$ if edge (v_i, v_j) separates two islands, otherwise $y_{i,j} = 1$).

$$\text{minimize} \quad \sum_{v_i \in \mathcal{L}} P_{LS,i} \tag{6.9a}$$

$$\text{subject to:} \quad \sum_{l=1}^{k} x_{i,l} = 1, \quad \forall v_i \in \mathcal{V} \tag{6.9b}$$

$$x_{i,l} = 1, \quad \forall v_i \in \mathcal{T}_l, \ \ l = 1,\ldots,k \tag{6.9c}$$

$$\begin{cases} z_{i,j,l} \le x_{i,l}, & \forall \left(v_i, v_j\right) \in \mathcal{E}, \ \ l = 1,\ldots,k \\ z_{i,j,l} \le x_{j,l}, & \forall \left(v_i, v_j\right) \in \mathcal{E}, \ \ l = 1,\ldots,k \\ z_{i,j,l} \ge x_{i,l} + x_{j,l} - 1, & \forall \left(v_i, v_j\right) \in \mathcal{E}, \ \ l = 1,\ldots,k \end{cases} \tag{6.9d}$$

$$y_{i,j} = \sum_{l=1}^{k} z_{i,j,l}, \quad \forall \left(v_i, v_j\right) \in \mathcal{E} \tag{6.9e}$$

$$\begin{cases} \sum\limits_{(v_i,v_j)\in\mathcal{E}} T_{i,j} - \sum\limits_{(v_j,v_i)\in\mathcal{E}} T_{j,i} = \sum\limits_{j\in\mathcal{V}} x_{j,l} - 1, & v_i = r_1,\ldots,r_k, \ \ l = 1,\ldots,k \\ \sum\limits_{(v_i,v_j)\in\mathcal{E}} T_{i,j} - \sum\limits_{(v_j,v_i)\in\mathcal{E}} T_{j,i} = -1, & v_i \in \mathcal{V} \setminus \{r_1,\ldots,r_k\} \\ T_{i,j} \le |\mathcal{V}| y_{i,j}, & \forall \left(v_i, v_j\right) \in \mathcal{E} \\ T_{i,j} \ge -|\mathcal{V}| y_{i,j}, & \forall \left(v_i, v_j\right) \in \mathcal{E} \end{cases} \tag{6.9f}$$

$$\begin{cases} P_{i,j} - B_{ij}\left(\delta_i - \delta_j\right) \le M_{ij}\left(1 - y_{i,j}\right), & \forall \left(v_i, v_j\right) \in \mathcal{E} \\ P_{i,j} - B_{ij}\left(\delta_i - \delta_j\right) \ge -M_{ij}\left(1 - y_{i,j}\right), & \forall \left(v_i, v_j\right) \in \mathcal{E} \\ P_{i,j} \le P_{i,j}^{max} y_{i,j}, & \forall \left(v_i, v_j\right) \in \mathcal{E} \\ P_{i,j} \ge -P_{i,j}^{max} y_{i,j}, & \forall \left(v_i, v_j\right) \in \mathcal{E} \\ \sum\limits_{(v_i,v_j)\in\mathcal{E}} P_{i,j} - \sum\limits_{(v_j,v_i)\in\mathcal{E}} P_{j,i} = \left(P_{G,i} - P_{GS,i}\right) - \left(P_{L,i} - P_{LS,i}\right), & \forall v_i \in \mathcal{V} \end{cases} \tag{6.9g}$$

Constraints (6.9b)–(6.9e) imply the partitioning status of each node (*partitioning constraints* (6.9b)–(6.9c)), and the fact that network edges running between nodes belonging to different partitions should be switched off (*switching constraints* (6.9d)–(6.9e)). However, the connectedness of graph partitititions defined by (6.9b)–(6.9e) is not ensured. To ensure the connectedness of the resulting islands, the artificial single commodity flow variables $T_{i,j}$ are often used [112, 196, 197]. To this end, one node in the $l-th$ island is selected as a *source node* producing $\sum_{j\in\mathcal{V}} x_{j,l} - 1$ units of an artificial commodity (i.e., the artificial commodity production in an island equals the total number of nodes in that island minus one). In (6.9f), the reference node of each group of terminal nodes serves as the source node. The remaining nodes in each island are sinks, each consuming one artificial commodity unit, which is reflected in the second equation of (6.9f). The third and forth equations in (6.9f) constrain artificial commodity flows not spread across open network branches, thus confining the flows from each reference node to its own island. If *connectivity constraints* (6.9f) can be satisfied, each island should be connected [112, 196, 197].

Finally, constraints (6.9g) link the split network topology expressed by the edge status variables $y_{i,j}$ to the DC OPF. The first two equations in (6.9g) represent the disjunctive constraints to equate the active power flow $P_{i,j}$ in edge $\{v_i, v_j\}$ to the status of that edge

encoded by $y_{i,j}$. If $y_{i,j} = 1$, the branch power flow is determined by the DC power flow expression $B_{ij}\left(\delta_i - \delta_j\right)$, where B_{ij} is the DC PF branch admittance, δ_i, and δ_j are the DC PF phase angles of nodes v_i and v_j. If $y_{i,j} = 0$, $P_{i,j}$ should be equal to zero due to (6.9g), and the phase difference $\left(\delta_i - \delta_j\right)$ becomes unconstrained because of the large positive *bigM constants* M_{ij}, which is a standard method of modeling logic through MILP [195]. As setting bigM constants to an unreasonably large value may lead to loose linear programming (LP) relaxations of MILP models, it is decided to set $M_{ij} = 2\pi B_{ij}$, which is large enough for study purposes (DC PF equations are also unreliable for large values of $\left|\delta_i - \delta_j\right|$). The third and fourth constraints in (6.9g) are constraining the MW flow variables $P_{i,j}$ to be zero if $y_{i,j} = 0$ and not to exceed the line limits if $y_{i,j} = 1$. The last equation in (6.9g) represents the active power balance for each node, where $P_{G,i}$ and $P_{L,i}$ are the MW generation and load levels at node v_i before splitting, and $P_{GS,i}$ and $P_{LS,i}$ are the generation reduction and load shedding amounts at node v_i after splitting. The bounds on $P_{GS,i}$ and $P_{LS,i}$ are implicit in (6.9g), and phase angles of reference buses $r_1, \ldots, r_k$ in each island are implicitly set to zero.

The formulation in (6.9) contains $k|\mathcal{V}| + (k+1)|\mathcal{E}|$ binary variables and $2|\mathcal{E}| + 3|\mathcal{V}|$ continuous variables that are related to DC OPF and artificial commodity flows. The total number of constraints (excluding constant bounds on variables) is equal to $3|\mathcal{V}| + (3k+7)|\mathcal{E}|$, which results in the number of constraints and decision variables growing linearly with the network size. Despite its compact size, the formulation in (6.9) may take a lot of time to solve to optimality as the network size grows due to its weak LP relaxation bounds as well as due to NP-hardness inherent to MILPs. However, it is often possible to reduce the MILP solution times by providing an good initial integer solution to the MILP solver, as it may help to improve the solver branching process by fathoming numerous search tree nodes with their LP relaxation values worse that that provided by the good initial integer solution.

In Table 6.4, the results of solving the MILP formulation in (6.9) are summarized. The test networks are taken from MATPOWER [81] and PST [83]. For MATPOWER networks, the missing electromechanical data for slow coherency analysis is generated from the standard generator data tables in [86], with the matching criterion being the closest nominal power rating. Furthermore, the tested generator groupings are produced by the combined slow coherency grouping algorithm in Section 5.3.2. For each test power network, the solutions of (6.9) involving 2–6 coherent generator groups have been tested. Testing the same group numbers for several very different power networks is only acceptable to evaluate the MILP solution performance, as the actual dominant electromechanical structures of each power network are individual. Since not all tested networks include information about branch power limits, the values of $P_{i,j}^{max}$ are uniformly set to twice the value of the maximal steady-state branch MW flow in the unsplitted network. The amount of load shedding $P_{LS,i}$ for every $v_i \in \mathcal{L}$ is assumed to be no less than zero and no more than the MW power consumption $P_{L,i}$ prior to ICI. The changes in generation $P_{GS,i}$ for every $v_i \in \mathcal{G}$ are assumed to be non-negative (i.e., no generation increases) and not exceeding 80% of the MW generation $P_{G,i}$ prior to ICI. The values of $P_{G,i}$ and $P_{L,i}$ are obtained from the AC power flow solution using the nominal network power flow data (see Appendix A).

The results in Table 6.4 were obtained by modeling (6.9) in the GAMS environment (GAMS v.26.1.0) with CPLEX being set as the MILP solver, on a PC with an Intel® Xeon® E5 3.70 GHz CPU and 16 Gb of RAM. The time limit to optimize (6.9) is set to 5 minutes, and the optimality gap tolerance is set to 1 %. In Table 6.9, the three types of results are contained:

1. Default CPLEX initialization with no initial solution provided to the solver, allowing CPLEX to perform the default model preprocessesing steps and then start the optimization from scratch.

2. Initialization from an initial solution satisfying constraints (6.9b)–(6.9f) that is obtained by applying Algorithms 6.1 and 3.3 on the active power flow graphs of the tested power networks. Algorithm 3.3 is used before Algorithm 6.1 to merge the unavailable branches and after Algorithm 6.1 to merge the resulting group trees. The actual constrained partitioning is obtained from the reduced power flow graph as described in Section 3.4.1 and [C1].

3. Initialization from an initial solution satisfying constraints (6.9b)–(6.9f) that is obtained by applying Algorithms 6.1, 3.3, and 3.2 on the active power flow graphs of the tested power networks. With this method, the solution obtained from the previous method using Algorithms 6.1, 3.3 is further improved by label propagation based graph cut refinement described in [138] and Algorithm 3.2.

The relevant solution outcomes in Table 6.4 are the objective value of the best integer solution LS^* in p.u., the MW cut value resulting from constrained graph partitioning cut_0 in p.u., the objective value of the initial integer solution obtained through constrained graph partitioning LS^0 in p.u., the time to solve (6.9) t in seconds, and the final optimality gap Gap between the best feasible integer solution and the best LP relaxation bound obtained by CPLEX. If an ICI test case could be solved to proven optimality within the prescribed time limit, the value of t should reflect this, and the value of Gap should be zero. In the opposite scenario, t should be equal to the prescribed time limit, and the value of Gap should be greater than 1 %.

Test network	k	Default CPLEX initialization			Initialization with Algorithms 6.1 and 3.3					Initialization with Algorithms 6.1, 3.3, and 3.2				
		LS^*	t, s	Gap, %	cut_0	LS^0	LS^*	t, s	Gap, %	cut_0	LS^0	LS^*	t, s	Gap, %
IEEE 68 bus test system, n=68, m=86	2	0.0	0.3	0.0	10.45	8.11	0.0	0.2	0.0	1.96	0.0	0.0	0.1	0.0
	3	0.0	0.3	0.0	3.57	1.14	0.0	0.4	0.0	3.56	1.14	0.0	0.4	0.0
	4	0.0	0.2	0.0	9.12	6.88	0.0	0.9	0.0	4.63	2.44	0.0	1.2	0.0
	5	1.5	0.6	0.0	19.5	13.9	1.5	0.4	0.0	4.73	2.79	1.5	0.6	0.0
NPCC 48 machine test system, n=140, m=233	2	0.0	0.4	0.0	17.85	0.17	0.0	0.4	0.0	5.88	0.0	0.0	0.1	0.0
	3	0.07	300	100	26.47	1.91	0.07	300	100	14.51	3.59	0.07	300	100
	4	8.52	300	100	41.63	30.93	8.52	300	100	20.05	8.52	8.52	300	100
	5	7.61	300	79.6	20.15	10.16	7.61	68	0.7	18.9	7.61	7.61	108	0.7
	6	10.07	300	84.6	33.72	*27.7*	10.07	300	20.4	21.44	*27.7*	10.07	300	40.3
IEEE 300 bus test system, n=300, m=411	2	0.0	189	0.0	31.48	11.34	0.0	59	0.0	5.23	4.35	0.0	165	0.0
	3	0.0	17	0.0	12.91	6.40	0.0	282	0.0	6.29	2.67	0.0	48	0.0
	4	1.37	300	100	28.44	16.86	0.0	222	0.0	16.18	9.04	0.0	77	0.0
	5	0.42	300	100	30.80	18.19	0.0	120	0.0	16.34	10.05	1.82	300	100
	6	2.14	300	100	32.01	13.36	0.02	300	100	18.11	10.05	5.76	300	100
Pegase 1354 bus test system, n=1354, m=1991	2	–	300	–	61.48	0.0	0.0	4.4	0.0	26.40	0.0	0.0	4.4	0.0
	3	0.0	12	0.0	37.59	0.0	0.0	0.6	0.0	36.47	0.0	0.0	0.6	0.0
	4	–	300	–	40.09	6.42	6.42	300	100	31.43	1.42	1.42	300	100
	5	–	300	–	45.19	6.42	6.42	300	100	37.01	1.42	1.42	300	100
	6	–	300	–	58.56	6.42	6.42	300	100	35.75	1.42	1.42	300	100

Table 6.4: Impact of initialization with Algorithms 6.1, 3.3, and 3.2 on solution of MILP-based ICI

The results in Table 6.4 allow to draw a number of conclusions:

1. The label propagation based graph cut refinement in Algorithm 3.2 is often very efficient at reducing constrained MW power flow cut values, which again confirms the conclusions from Figures 3.5–3.6.

2. Lower MW power flow cuts mostly result in noticeably lower initial load shedding solutions.

3. A low initial load shedding may still result in longer solution times. That is, low MW power cuts often lead to good local optima of (6.9), but this factor alone is not always enough to ensure a rapid convergence to the global optimum when tackling (6.9) with sophisticated commercial MILP optimizers (e.g., CPLEX, Gurobi [87]).

4. Satisfying constraints (6.9b)–(6.9f) through constrained graph partitioning that minimizes the MW power flow cut is not guaranteed to satisfy the DC OPF constraints (6.9g). For example, the initial feasible solution for the NPCC 48 machine test system could only be obtained through the CPLEX heuristics that "repaired" the provided initial solution.

5. Nevertheless, as the network size grows and the efficiency of MILP-based ICI declines, the ability of Algorithm 6.1 to quickly find a good feasible solution becomes more valuable. For example, it was not possible to find any feasible solution for the Pegase 1354 bus test network for 2, 4, 5, and 6 islands without providing an initial constrained partitioning solution.

To conclude this case study, some modeling and algorithmic limitations should be mentioned. The ICI formulation in (6.9) exactly models the graph partitioning constraints (i.e., the node grouping constraints and island connectedness). The island power balance constraints and branch power flow limits are modeled by the *approximate* linear DC power flow relationships (6.9g), while voltage magnitudes and reactive powers are completely neglected because of the nature of DC OPF equations (6.9g). The graph cut refinement Algorithm 3.2 needs to be modified to fix the merged generator group trees at their respective islands. For the experiments in Table 6.4, this change was not implemented, but the merged nodes representing the group trees were never moved away from their islands, possibly because of the large weight of all the edges in the reduced graph that were connected to them.

6.5. CONCLUSIONS

This chapter considered the problem of graph partitioning subject to node grouping constraints, which directly translates to coherent generator grouping constants of ICI [55] and generator cranking groups constraints of PPSR [45, 54], but can also be instrumental to some other WAMPAC strategies. The major focus was on approximately solving this NP-hard constrained graph partitioning problem in polynomial time, while aiming at minimizing the number of misplaced terminal nodes, which is precisely the subject of research question IV (see Section 1.2.3).

The overall chapter structure was subdivided into three parts. The first part described a transformation of power system similarities (e.g., branch admittances, power flows, or various sensitivities) into distances that can also incorporate the node grouping constraints. This transformation is based on FCSC, which extends traditional spectral clustering to minimize both the weighted edge cut between the partitions and the number of violated pairwise grouping constraints (i.e., must-link and cannot-link constraints), with a special FCSC tuning parameter determining the relative importance of each objective. Therefore, the geometric node coordinates returned by FCSC are biased towards the fulfilment of input grouping constraints. As illustrated by Figure 6.2, this bias can be made significant by appropriately setting the FCSC tuning parameter α'.

The FCSC algorithm can be used to redefine the edge weights of the power network topology graph $\mathscr{G}$ based on the distances between the edges' end nodes induced by the FCSC graph embedding. The second part of this chapter concentrated on the newly proposed sequential tree growing algorithm to approximately solve the NP-hard packing Steiner trees problem. This algorithm is aiming to find the k disjoint group trees, each spanning the buses of a single terminal nodes' group, and it includes the measures to coordinate the assignment of buses to multiple trees. In particular, the newly computed paths to interconnect a terminal nodes group are constrained not to pass through the nodes that are already assigned to other group trees or through the nodes that are possibly required to interconnect other group trees. Because of these measures, the proposed sequential tree growing algorithm was shown to consistently outperform the two state-of-the-art alternatives [126] and [91] in terms of satisfying the node grouping constraints while demonstrating an acceptable computational performance for power networks consisting of several thousands of nodes.

The final part investigated the use of the earlier proposed polynomial-time partitioning algorithm to provide initial solutions for MILP-based ICI. It was shown that lower MW power flow cuts often result in lower initial values of total load shedding, but not necessarily in shorter solution times. The active power flow cuts were decreased (often significantly) from their initial values resulting from Algorithm 6.1 by applying the label propagation graph cut refinement in Algorithm 3.2, thus highlighting the practical value of the post-processing algorithms in Chapter 3. It was also observed that the value of good initial solutions found through polynomial-time constrained graph partitioning increases as the network size grows and solving MILPs becomes more difficult (due to more discrete variables, larger LP relaxation size etc.).

Although FCSC described in Section 6.2 is instrumental at converting power flow similarities into distances, alternative graph distance metrics (e.g., based on branch impedances or commute-time distances [30]) can be used with Algorithm 6.1 as well depending on the studied application.

7

CONCLUSIONS AND RECOMMENDATIONS

The main topic of this thesis is the discovery of power system structure resulting in partitioning of power networks into internally cohesive and well-separated areas. The research methodology predominantly relies on graph theory, graph algorithms, and graph partitioning, with spectral graph theory being the major theoretical framework.

From the power engineering perspective, this thesis considers clustering of static and dynamic power system graphs. Static graphs are based on data derived from static power system analysis, including branch admittance graphs, power flow graphs, and voltage sensitivity graphs. Dynamic graphs reflect the power system dynamic properties, and they arise in this thesis to model slow coherency.

From the algorithmic perspective, this thesis studies two types of graph partitioning problems, which are termed as constrained and unconstrained graph partitioning. Constrained graph partitioning deals with partitioning a network subject to node grouping constraints, and it has been shown to be related to the packing Steiner trees problem in combinatorial optimization. The power system applications involving constrained graph partitioning include ICI and PPSR. Unconstrained graph partitioning aims to partition a network into a set of well-separated clusters without constraints or assumptions on cluster assignment of any particular node. In this thesis, it is shown that AZD for SVC and generator slow coherency can be successfully resolved through unconstrained graph partitioning.

The final outcome of this thesis is a generic framework for constrained and unconstrained partitioning of power system graphs. The subsequent sections describe the main research contributions of this framework, answer the research questions raised in Section 1.2.3, and provide suggestions for future work.

7.1. RESEARCH CONTRIBUTIONS

This section aims to provide an overview of the research contributions of this thesis.

7.1.1. DETERMINING THE OPTIMAL NUMBER OF CLUSTERS

Several applications in power system control and operations require the power system to be subdivided into a number of zones or areas. Examples of such applications include zonal pricing, DSA, SVC, PPSR, and controlled network separation. One important issue arising in such applications is the choice of the number of areas to partition the network [46, 94]. Many existing approaches to infer the number of areas depend on an underlying clustering technique (e.g., [39, 46, 52, 173]), thus sharing its biases and drawbacks. Another popular method to estimate the number of clusters is based on eigengaps in graph spectra (e.g., [34, 76, 94, 117]), but it may be ambiguous if the studied graph does not possess a distinct clustering structure [129].

In this thesis, a procedure to detect good choices for the number of clusters was proposed that is not based on a clustering algorithm and more informative than the spectral eigengap heuristic. This procedure is based on aligning the graph spectral embedding of varying dimensionality with the canonical coordinate axes and choosing the number of clusters as the dimension of the spectral embedding that allows for the best alignment. The spectral embedding is as an inherent graph property that depends on the graph interconnection structure and the strength of each connection. By including the coordinates of every graph node, the aligned spectral embedding is directly related to the graph clustering structure, while there is no such direct relationship for spectral eigengaps.

The initial algorithm for aligning spectral embeddings with the standard basis was formulated in Chapter 4. In Chapter 5, this algorithm was extended to accommodate for the specifics of the slow coherency grouping problem. The SVC AZD case studies in Section 5.2.4 and the slow coherency case studies in Section 5.3 demonstrated that the proposed approach to select the number of areas discovers area structures that are both empirically and numerically meaningful.

7.1.2. GRAPH PARTITIONING USING ALIGNED SPECTRAL EMBEDDING

To further utilize the valuable information contained in aligned spectral embeddings, an efficient k-way spectral partitioning algorithm was formulated in Chapter 4. This algorithm separately considers each coordinate of the aligned spectral embedding to estimate the k cluster cores around which the final partitions should be formed. One possible approach to partition a network around the cluster cores was formulated in Section 4.4. In Section 4.5, this approach was successfully tested on the branch admittance graphs of two large-scale power networks from MATPOWER. The test results showed that the proposed graph k-way spectral partitioning algorithm outperformed three other benchmark algorithms, and often by a large margin. In Chapter 5, the k-way partitioning algorithm of Chapter 4 was additionally extended to accommodate for the specifics of the slow coherency grouping problem.

7.1.3. GRAPH PARTITIONING BASED ON SEQUENTIAL TREE GROWING

Constrained graph partitioning with respect to node grouping constraints was another important topic of this thesis. It was discussed in this thesis that the constrained graph partitioning problem is NP-hard due to its close relation to the packing Steiner trees problem. Due to the inherent computational hardness of computing the exact solution, polynomial-time heuristic algorithms that could quickly provide a good feasible solu-

tion are of special interest. As it is presently not possible to always attain feasibility when tackling an NP-hard problem with a polynomial-time algorithm, it may be desirable to at least minimize the infeasibility of the solution returned by the heuristic approach. For the discussed constrained graph partitioning problem, the minimal infeasibility requirement translates into the minimal number of violated node grouping constraints, the metric that was introduced in Section 2.7.

A new constrained graph partitioning approach proposed in Chapter 6 was designed along the above-mentioned ideas. It attempts to construct k Steiner trees, each spanning its own group of terminal nodes. The tree construction process proceeds by connecting one terminal node at a time; the connections with highest chances not to violate any node grouping constraints receive the highest priority, while connections that are identified to potentially violate some node grouping constraints are shifted to the end of the priority queue. The comparison with two state-of-the-art alternatives was performed in Section 6.3.2, and the test results showed that the proposed algorithm consistently violates less node grouping constraints, and often by a large margin. As the next step, the proposed sequential tree growing algorithm was used as the initialization heuristic for a MILP-based ICI formulation. This case study largely confirmed the assumption that constrained graph partitioning solutions with a low power flow disruption cause a lower initial load shedding of MILP-based ICI.

7.1.4. PRE- AND POSTPROCESSING FOR GRAPH PARTITIONING

In Chapter 3, a number of auxiliary measures were introduced with the goal to resolve some of the inherent graph partitioning drawbacks and to better adapt graph partitioning to power system specific problems. In particular, graph outlier detection and the graph cluster connectedness heuristic largely serve the first goal, while graph cut refinement and graph reductions largely serve the second goal.

The proposed graph cluster connectedness heuristic in Section 3.3.1 was inspired by the well-known graph partitioning library METIS [90], and it was used throughout the thesis to enforce connected graph partitions for various benchmark algorithms that do not support this requirement. The proposed graph outlier detection algorithm in Section 3.4.2 was shown to prevent some existing graph partitioning algorithms from returning very small clusters. Graph reductions were used previously in [58]. The approach described in Section 3.4.1 differs from [58], but its major goal is the same (i.e., the enforcement of ICI-related node grouping constraints). Finally, the graph cut refinement in Section 3.3.2 was adopted from [138], but its use to reduce the initial load shedding value of MILP-based ICI can be considered as another contribution of this thesis.

7.1.5. NEW ALGORITHMS FOR SVC AZD AND SLOW COHERENCY

The clustering framework based on aligning spectral embeddings with the canonical coordinate axes (Contributions 7.1.1 and 7.1.2) was successfully applied to devise new techniques for SVC AZD division and slow coherency. In the case of AZD, the framework introduced in Chapter 4 was successfully applied to identify the voltage control zones for the IEEE 39 and 68 bus test power systems. The SVC case studies on those test power systems showed a significant voltage error reduction compared both with the scenarios with no SVC and the scenario with SVC controlling an alternative set of pilot buses.

In Section 5.3.2, the analogy between power system slow coherency [34] and spectral clustering was first highlighted. Although the classical slow coherency problem (5.10) can be linked to spectral graph theory, it is different from the classical normalized spectral clustering studied in Chapter 4. In particular, the eigenvalues of $\mathbf{M}^{-1}\mathbf{K}$ are not restricted to $[0,-2]$, the generic expansion (5.12) is not bounded in $[0,1]$, and matrices $\mathbf{K}$ or $\mathbf{K^B}$ represent a complete graph. Thus, the previously introduced spectral clustering algorithms of Chapter 4 were extended to accommodate some facts known from classic slow coherency theory [33, 34]. The case studies of Section 5.3.5 showed that spectral clustering techniques from both Chapters 4 and 5 perform well for the slow coherency grouping task, especially when the matrix of synchronizing torque coefficients is assumed to be symmetric. However, the extensions added in Chapter 5 resulted in a better performance of the slow coherency grouping algorithm from Section 5.3.2 for a number of test cases.

7.2. ANSWERS TO RESEARCH QUESTIONS

I. What are the implications of high area cohesiveness and separation on the efficiency of WAMPAC applications and how can they be assessed?

This question expresses the main motivation to research the topics of graph clustering and graph partitioning in the context of power systems. To answer it, a spectral clustering framework that is novel to electric power system research was proposed in Chapter 4 and extended in Chapter 5. In this thesis, area separation was mostly assessed through normalized cuts (2.7), (5.11) and cluster expansion ratios (2.23), (5.12). Area cohesiveness was assessed indirectly by choosing area divisions featuring a high alignment cost (4.2), (4.5). This approach to area cohesiveness was illustrated in the SVC AZD case study of Section 5.2.4 involving the IEEE 39 bus test power system. In that case study, a line trip caused a significant reduction of the internal connectivity of SVC Zone 5. As the result, the spectral clustering alignment cost of the currently chosen VCS division significantly increased, signaling the decrease in internal cohesiveness of Zone 5 and the need to choose a different number of zones corresponding to a lower alignment cost. Thus, it appears that a low spectral clustering alignment cost achieves a trade-off between area separation and area cohesiveness for a desired clustering resolution level (i.e., the desired range of k). Choosing the number of VCZs for SVC based on local minima of spectral clustering alignment cost was shown to result in high-quality voltage regulation in Section 5.2.4. As SVC requires evaluating a set of pilot buses, but not VCZs as whole, assessing the impact of low cluster expansions and normalized cuts on voltage regulation quality was problematic. However, the slow coherency case studies in Section 5.3.5 have demonstrated the positive impact of low expansion ratios and normalized cuts on generator slow coherency. Thus, pursuing high area cohesiveness and separation was shown to have a positive influence on two area-based control applications; it is also likely to be beneficial for some other existing and prospective WAMPAC applications.

II. How to achieve a greater control over the clustering results (e.g., to avoid very small areas), while not compromising the computational efficiency?

This question was motivated by the noticeable deficiencies of off-the-shelf graph partitioning approaches when directly applied to problems in power systems. All techniques proposed in this thesis were devised with this goal in mind. In Chapter 3, four auxiliary algorithms were developed to circumvent some of the drawbacks present in many off-the-shelf graph partitioning algorithms. The summary of this work can be found in Contribution 7.1.4. Some other aspects of this problem are discussed in the answers to research questions III and VI.

> *III. How to ensure area connectivity when applying clustering algorithms to identify areas in power networks?*

This question was motivated by the common drawback of many off-the-shelf graph partitioning algorithms, which are prone to return disconnected graph clusters. In Chapter 3, a graph cluster connectedness heuristic was proposed that allowed to "repair" disconnected graph clusters, while optimizing some graph partitioning quality metric (maximal cluster expansion or normalized cut were assumed as such metric in Section 3.3.1). In Chapter 4, area connectivity was ensured by deliberately enforcing the graph cluster cores to be connected and then using minimum s-t cuts to finalize the partitioning. In Chapter 6, a separate Steiner tree was constructed for each group of terminal nodes to ensure the connectivity of these nodes – this principle was briefly mentioned in Section 3.4.1 before being extended in Chapter 6.

> *IV. How to achieve a high degree of satisfaction of node grouping constraints by using constrained clustering algorithms while ensuring area connectivity?*

This question was motivated by the important ICI requirement to only include coherent generators into each island, while ensuring the connectedness of each island. The proposed solution is based on the analogy between constrained graph partitioning with respect to node grouping constraints and the packing Steiner trees problem in combinatorial optimization [77, 78]. The sequential tree growing algorithm was proposed in Chapter 6 to solve this problem, while minimizing the number of unsatisfied node grouping constraints. The brief summary of this algorithm was given in Contribution 7.1.3, with its full description and test results being available in Section 6.3.

> *V. How to determine an optimal number of areas for various power system analysis and WAMPAC applications?*

For WAMPAC applications requiring a complete partitioning of the power network into a number of areas, an algorithm to determine good choices for the number of areas was proposed in Chapter 4. The summary of this algorithm is provided in Contribution 7.1.1. For this method to be applicable, the power system area identification problem should be representable as a graph partitioning problem with an underlying similarity graph model. Thus, a number of area identification problems (e.g., identification of critical VCAs [36, 117]) cannot be directly tackled with the proposed solution due to their specific nature. However, the proposed method to identify an optimal number of clusters can be potentially applied to power system grouping problems that are not directly related to area identification, but can be represented with similarity graphs (e.g., the contingency clustering problem [117]).

VI. How to enhance clustering algorithms to satisfy a larger number of power-system related constraints in a timely manner?

This question further develops research question II by asking for the new methods to adapt clustering algorithms to power system related problems. In this thesis, the following constrained clustering issues were resolved:

- Several strategies to tackle the area connectivity issue were proposed (see the answer to research question III).
- An efficient polynomial-time algorithm to resolve node grouping constraints was proposed (see the answer to research question IV).
- Two approaches to avoid too small areas were proposed (the outlier mining algorithm in Section 3.4.2 and the avoidance of too small cluster cores in Section 4.4).
- Encoding of node grouping constraints through graph reductions in Section 3.4.1.

7.3. RECOMMENDATIONS FOR FUTURE WORK

7.3.1. EXTENSION TO OTHER GROUPING PROBLEMS IN POWER SYSTEMS

In this thesis, a spectral clustering based grouping framework was proposed and applied to AZD for SVC and generator slow coherency. These two applications are relevant for area-based control and protection are require subdividing a power system into a number of areas. Related applications include area-based online DSA [39, 40], area-based PMU placement, damping of inter-area oscillations [35, 198], wide-area voltage protection (WAP) etc. Certain applications in power system operations (e.g., [30, 163]) could also benefit from the proposed clustering framework, especially in terms of estimating the suitable number of clusters.

Additionally, there is a large number of use cases of power network partitioning and general clustering of power system data that are not directly related to WAMPAC. For example, the convergence characteristics of distributed optimization algorithms (e.g., alternating current (AC) OPF [120]) could be improved by optimally partitioning the power system into subsystems. Other possible use cases include contingency clustering [117], search space reduction for large-scale optimization [13], reduction of large-scale sets of real-time data collected by transmission system operators (TSOs) and others.

7.3.2. IMPACT OF WIND AND SOLAR GENERATION ON SLOW COHERENCY

The slow coherency grouping algorithm proposed in Chapter 5 was based on classic slow coherency theory for synchronous machines. Although some substantial improvements over the classic slow coherency algorithm were demonstrated, the used modeling framework was still the same as in 1980s–1990s (i.e., accounting only for large conventional synchronous generators). With the current trends for increasing the share of renewable wind and solar power plants, this classic framework to model rotor angle dynamics needs to be updated to adequately represent the impact of renewable generation. It is desirable for a new model to be conceptually similar to the classic one for the existing methods to remain applicable.

7.3.3. MEASUREMENT-BASED GENERATOR COHERENCY

A number of works has been published on clustering generator signals estimated by PMUs [49, 50, 164]. Some of these methods form a similarity matrix from the signal data and subsequently retrieve the generator groups from it by applying a clustering technique. Although this thesis did not tackle the problem of building a generator similarity matrix from measurements, once such matrix is available, the clustering framework of Chapter 4 could be used to analyze it and retrieve the generator groups.

7.3.4. INCLUSION OF AC POWER FLOW CONSIDERATIONS INTO ICI

In this thesis, the ICI problem was tackled in a simplified way by only considering generator coherency and active power flows (i.e., constrained graph partitioning minimizing active power flow cuts and MILP-based DC OPF ICI). However, it is known that optimal solutions obtained with a MILP-based DC OPF ICI model may be AC infeasible [113]. The AC infeasibility of network splitting solutions or significant violations of bus voltage bounds may be due to a global reactive power imbalance in an island or due to local reactive power mismatches. To alleviate the first problem, it is conceivable to adjust the constrained graph partitioning algorithms to minimize the reactive power flow cut, possibly in combination with the MW power flow cut as it was done in [107]. Although such measures may be useful for quickly finding good initial splitting solutions, they may be insufficient to resolve certain peculiar situations (e.g., those involving local imbalances of reactive power). Therefore, it is desirable to develop computationally efficient exact methods to solve power system splitting problems that include AC power flow constraints.

7.3.5. WHEN TO ISLAND

This thesis was exclusively concerned with the problem *"Where to island?"*, i.e. which set of transmission lines to open for a stable and economical operation of the separated power network. The equally important problem of *"When to island?"*, i.e. when to activate ICI to prevent an imminent wide-area instability, remained completely out of scope and could be considered as another future research topic.

ACKNOWLEDGEMENTS

This thesis summarizes some of the outcomes of the project "PMU Supported Frequency-Based Corrective Control of Future Power Systems" that is a part of the Uncertainty Reduction in Smart Energy Systems (URSES) program. I would like to express my gratitude to all of the program officers at the Dutch Research Council (NWO) for their support and for making this research possible.

Foremost, I would like to thank my promotors Marjan Popov and Mart van der Meijden for trusting me to conduct this Ph.D. research. Their knowledge, guidance, and support have contributed a lot to the outcomes of this thesis. Mart, it was a great experience to communicate with you and to grasp some of your wisdom and positive attitude. Marjan, thank you for being a great daily supervisor ("not-a-boss"), who is sincerely interested in the success of his students. I truly appreciate your patience and support in all difficult and uncertain situations that occurred during my Ph.D. journey.

I am very grateful to the members of my doctoral examination committee for reviewing and evaluating my thesis. Their comments were very valuable for the improvement of the quality of this dissertation. A special thanks goes to Professor Vladimir Terzija from the University of Manchester, who was always keen to support my research initiatives as an external project member. It was always pleasure to talk to you, Vladimir, and I learned a lot from our insightful discussions about the research and beyond. Another special thanks goes to Dr. Jairo Quirós-Tortós from the University of Costa Rica for helping me in the early stages of the project. Thank you, Jairo, for all the pieces of advice, data, and models that you provided to advance my research. One more special thanks is for Jorrit Bos from TenneT TSO for being keenly interested in the work of our project. The final special thanks goes to Dr. Christian Schulz, who was very kind to provide the modified versions of some routines from the KaHIP graph partitioning library.

To a large degree, the achieved results were made possible by the friendly and collaborative environment inside of the Intelligent Electric Power Grids (IEPG) research group, which is a part of the Electrical Sustainable Energy department at Delft University of Technology. In this regard, I would like to thank Peter Palensky for truly being an enthusiastic leader (not a boss) of the IEPG group, Ellen, Sharmila, Ilona for being the best-ever secretaries, and Dr. Jose Rueda Torres for his constant encouragement and help with DIgSILENT PowerFactory. It is extremely important to mention my previous and current officemates Dr. Arjen, Matija, Aleksandar and especially Arkadio. Thank you, guys, for the nice working atmosphere, open discussions, good jokes, and support in all kinds of situations. I was also lucky to be a colleague of many great fellow PhD students and postdocs, including Dr. Romain, Nakisa, Isidora, Dr. Bart, Dr. Mario, Dr. Hossein, Claudio, Dr. Swasti, Rishabh, Dr. Lian, Kaikai, Arun, Vinay, Digvijay, Aihui, Hazem, Chenguang, Umer, Roland, Dr. Jose, Dr. Ebrahim, Dr. Ayman, Dr. Elyas, Dr. Zameer (I hope I didn't forget anyone!). Thank you all for being a part of my PhD journey!

I am also grateful to the Russian-speaking community in Delft and its surroundings (Dima, Misha, Dima, Norman, Andrei, Yura, Makar, Artur, Nick, Katya, Polina, Yana, Sasha, Pasha et al.) for many good moments we shared together, for the moral support in difficult situations, and for the sense of community and friendship that unites us. In general I can say that all other people whom I met during my PhD (be it students, professors, or normal citizens) were good, so everyone whom I met probably helped me at least a tiny bit.

Finally, my family always was my main motivation to reach goals and achieve good results. Needless to say that I owe my family the most. When my PhD project was already coming to an end, I got shocking news that my dear mother Natalia suddenly passed away. This was a very sad period in my life, but it is not possible to turn the time back. I am definitely grateful to my mother the most for what she has done for me, and although she is no longer with us, she will always remain in my heart.

Ilya Tyuryukanov
Delft, March 2020

A

TEST POWER NETWORKS

This thesis makes use of multiple test power networks to provide appropriate and sufficiently diverse test cases for the proposed algorithms and their applications. While it could be reasonable to include the data for smaller test power networks (e.g., the IEEE 39 bus test system or the IEEE 68 bus test system), large-scale test power networks from MATPOWER involve too much data to be included into a text document. The test power networks that are relevant for this thesis are listed in Table A.1 together with the sources of their data, but without including the data explicitly. Unless otherwise stated, the nominal power network data contained in the references below is used in the case studies.

Test network	Data source	Use in the thesis
IEEE 39 bus test power network	Appendix A in [180]	Sections 3.4.1, 3.4.2; Section 4.3.2; Section 5.2.4; Section 6.2;
IEEE 68 bus test power network	Appendix A in [141]	Section 3.3; Section 5.2.4; Section 6.4;
NPCC 48 machine test power network	file `datanp48.mat` in Power System Toolbox [83]	Section 5.3.5; Sections 6.3.2, 6.4;
IEEE 145 bus test power network	file `data50m.m` in Power System Toolbox [83]	Section 5.3.5;
IEEE 300 bus test power network	file `case300.m` in MATPOWER [81]	Sections 6.3.2, 6.4;
PEGASE 1354 bus test power network	file `case1354pegase.m` in MATPOWER [81]	Section 3.4.2; Section 4.5; Sections 6.3.2, 6.4;
PEGASE 2869 bus test power network	file `case2869pegase.m` in MATPOWER [81]	Section 3.4.2; Section 4.5; Section 6.3.2;
Polish 2383 bus test power network	file `case2383wp.m` in MATPOWER [81]	Section 3.4.2; Section 4.5;

Table A.1: Data sources for used test power networks

For the power network models from MATPOWER, only power flow data is available. For studies involving dynamic generator data (e.g., to build the electromechanical power system model), the missing dynamic data is generated from the standard generator data tables in [86], with the matching criterion being the closest nominal power rating.

B

GRADIENT DESCENT BASED EIGENVECTOR ALIGNMENT

To minimize the cost function in (4.5), the authors of [129] used a gradient descent (GD) based optimization scheme in which the optimized orthogonal matrix $\mathbf{R}$ was represented as a series of Givens rotations [128], with Givens angles serving as the optimization variables. The expression for $\mathbf{R}$ as the function of Givens angles can be written as:

$$\mathbf{R}(\theta_{1,2},\ldots,\theta_{k-1,k}) = \prod_{i=1}^{k-1}\prod_{j=i+1}^{k} \mathbf{G_{i,j}}(\theta_{ij}) \tag{B.1}$$

where $\mathbf{G_{i,j}}(\theta_{ij})$ is a counterclockwise rotation matrix by angle θ_{ij} in the plane spanned by i and j coordinate axes. The angles θ_{ij} are the optimization variables in (4.5), and $\boldsymbol{\theta} = (\theta_{1,2},\ldots,\theta_{k-1,k})$ is their vector representation.

The classical update rule for the batch GD algorithm to minimize the cost function in (4.5) is given as follows [71]:

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \alpha\nabla J(\boldsymbol{\theta}_{t-1}) \tag{B.2}$$

where $\boldsymbol{\theta}_t$ is the argument vector at iteration t, $\nabla J(\boldsymbol{\theta}_{t-1})$ is the cost function gradient at $\boldsymbol{\theta}_{t-1}$, and α is the learning rate parameter. The argument vector $\boldsymbol{\theta}$ is updated at every iteration until the objective function plateaus at some value or the maximal number of iterations is reached.

For many practical problems involving large and high-dimensional datasets, the update rule in (B.2) results in a slow rate of convergence. To circumvent this issue, several GD optimization variants are used by the machine learning community [199]. The experiments in Chapter 4 are realized by using the GD variant known as *Nesterov acceler-*

ated gradient [200] given by its description in [199]. Thus, the used GD update rule is as follows:

$$\boldsymbol{p}_t = \gamma \boldsymbol{p}_{t-1} + \alpha \nabla J(\boldsymbol{\theta}_{t-1} - \gamma \boldsymbol{p}_{t-1}) \tag{B.3a}$$

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \boldsymbol{p}_t \tag{B.3b}$$

where $\boldsymbol{p}_t$ is the Nesterov momentum term at iteration t, and γ is the damping coefficient. The momentum principle gives the GD algorithm (B.3) a cumulative memory about the past steps, and its special form as given in Equation (B.3a) works according to the predictor and corrector principle (more details can be found in [199, 201]). The formulation in (B.3) has been compared to several other GD algorithms explained in [199], and it has been selected due to the observed superior performance in minimizing cost (4.5) (both in terms of run time and objective values) as well as the conceptual simplicity.

The last ingredient to optimize (4.5) using GD is the analytical expression for the gradient of the objective function. While numerically estimated gradients can be used instead, the optimization performance may suffer due to the assumptions involved. To derive the analytical expression for (4.5), the notational conventions from [129] are going to be followed. First, the Givens rotation matrices are re-labeled as $\mathbf{G_l}(\theta_l) = \mathbf{G_{i,j}}(\theta_{ij})$, where each pair (i, j) indicating a Givens rotation angle becomes re-mapped to an integer l with $l = 1, \ldots, K$ and $K = k(k-1)/2$. Next, the matrix defining the Givens rotation involving the angles $\theta_a, \ldots, \theta_b$ is denoted as $\mathbf{R}_{(a,b)} = \mathbf{G_a}(\theta_a)\mathbf{G_{a+1}}(\theta_{a+1}) \cdots \mathbf{G_b}(\theta_b)$. Next, define $\mathbf{Z} = \mathbf{UR}$ and $\mathbf{A}^{(k)} = \frac{\partial \mathbf{Z}}{\partial \theta_k}$ with $A^{(k)}_{ij} = \frac{\partial Z_{ij}}{\partial \theta_k}$. Using the introduced matrix definitions, it is possible to obtain $\mathbf{A}^{(k)} = \mathbf{UR}_{(1,k-1)}\mathbf{R}_{(k+1,K)}\frac{\partial \mathbf{R}_{(k,k)}}{\partial \theta_k}$. With the above conventions, the gradient $\nabla J(\boldsymbol{\theta})$ can be derived element-wise:

$$\begin{aligned}
\frac{\partial J}{\partial \theta_k} &= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}\frac{\partial}{\partial \theta_k}\left(Z_{ij}^2 \frac{1}{M_i^2}\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k} 2\frac{Z_{ij}}{M_i^2}\frac{\partial Z_{ij}}{\partial \theta_k} - 2\frac{Z_{ij}^2}{M_i^3}\frac{\partial M_i}{\partial \theta_k}
\end{aligned}$$

to yield the following formula:

$$\frac{\partial J}{\partial \theta_k} = \frac{2}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}\frac{Z_{ij}A_{ij}^{(k)}}{M_i^2} - \frac{Z_{ij}^2 A_{ij_{mi}}^{(k)}}{M_i^3} \tag{B.4}$$

where $j_{mi} = \text{argmax}_j Z_{ij}$ (i.e., $M_i = Z_{ij_{mi}}$).

C

CSVC Control Law

This appendix contains a practical description of the CSVC control law proposed in [96] that is used in Section 5.2 to evaluate various sets of pilot buses. The goal of the CSVC control law [96] is to maintain the voltages at pilot buses close to the reference values while simultaneously balancing the reactive loading levels of control generators, which was expressed in [96] as a quadratic programming problem:

$$
\begin{aligned}
\text{minimize}_{\boldsymbol{\Delta V_C}} \quad & \left\| \alpha \boldsymbol{\Delta V_P} - \left[\frac{\partial V_P}{\partial V_C} \right] \boldsymbol{\Delta V_C} \right\|_2^2 + \beta \left\| \left(\text{diag}(\overline{\boldsymbol{Q}}_{\boldsymbol{C}}) \right)^{-1} \left(\boldsymbol{Q_C} + \left[\frac{\partial Q_C}{\partial V_C} \right] \boldsymbol{\Delta V_C} \right) - \right. \\
& \left. \left(\text{diag}(\overline{\boldsymbol{Q}}_{\boldsymbol{C}}^{\Downarrow}) \right)^{-1} \left(\boldsymbol{Q}_{\boldsymbol{C}}^{\Downarrow} + \left[\frac{\partial Q_C}{\partial V_C} \right]^{\Downarrow} \boldsymbol{\Delta V_C} \right) \right\|_2^2 \\
\text{subject to} \quad & \underline{\boldsymbol{V}}_{\boldsymbol{C}} \leq \boldsymbol{V_C} + \boldsymbol{\Delta V_C} \leq \overline{\boldsymbol{V}}_{\boldsymbol{C}} \\
& \underline{\boldsymbol{V}}_{\boldsymbol{PQ}} \leq \boldsymbol{V_{PQ}} + \left[\frac{\partial V_{PQ}}{\partial V_C} \right] \boldsymbol{\Delta V_C} \leq \overline{\boldsymbol{V}}_{\boldsymbol{PQ}} \\
& \underline{\boldsymbol{Q}}_{\boldsymbol{PV}} \leq \boldsymbol{Q_{PV}} + \left[\frac{\partial Q_{PV}}{\partial V_C} \right] \boldsymbol{\Delta V_C} \leq \overline{\boldsymbol{Q}}_{\boldsymbol{PV}} \\
& \underline{\boldsymbol{Q}}_{\boldsymbol{C}} \leq \boldsymbol{Q_C} + \left[\frac{\partial Q_C}{\partial V_C} \right] \boldsymbol{\Delta V_C} \leq \overline{\boldsymbol{Q}}_{\boldsymbol{C}}
\end{aligned}
\tag{C.1}
$$

where underlining and overlining respectively denote lower and upper equipment limits, $\boldsymbol{\Delta V_C}$ is the SVC control generator terminal voltage correction vector (optimization variable vector), $\boldsymbol{\Delta V_P}$ is the vector of voltage deviations at pilot buses, $\left[\frac{\partial V_P}{\partial V_C} \right]$ is the sensitivity matrix relating $\boldsymbol{\Delta V_C}$ and $\boldsymbol{\Delta V_P}$, $\boldsymbol{\Delta Q_C}$ is the vector of control generator reactive power outputs, $\left[\frac{\partial Q_C}{\partial V_C} \right]$ is the sensitivity matrix relating $\boldsymbol{\Delta V_C}$ and $\boldsymbol{\Delta Q_C}$, $\boldsymbol{Q}_{\boldsymbol{C}}^{\Downarrow}$ is the vector $\boldsymbol{Q_C}$ circularly shifted one row down, $\left[\frac{\partial Q_C}{\partial V_C} \right]^{\Downarrow}$ is the matrix $\left[\frac{\partial Q_C}{\partial V_C} \right]$ circularly shifted one row down, $\boldsymbol{V_{PQ}}$ is the vector of voltages at PQ buses not selected as pilot buses, $\left[\frac{\partial V_{PQ}}{\partial V_C} \right]$ is the sensitivity matrix relating $\boldsymbol{\Delta V_C}$ and $\boldsymbol{\Delta V_{PQ}}$, $\boldsymbol{Q_{PV}}$ is the vector of reactive power outputs

of non-controlling generators (fixed PV buses), $\left[\frac{\partial Q_{PV}}{\partial V_C}\right]$ is the sensitivity matrix relating $\boldsymbol{\Delta V_C}$ and $\boldsymbol{\Delta Q_{PV}}$, β is the balancing parameter between the two CSVC objectives, and α is the weight parameter to compute the CSVC control action in a number of small steps spread over time.

In the above description, the matrices $\left[\frac{\partial V_P}{\partial V_C}\right]$ and $\left[\frac{\partial V_{PQ}}{\partial V_C}\right]$ can be obtained from the matrix **S** in (5.6) by taking the submatrices of **S** corresponding to the sensitivity of respectively $\boldsymbol{\Delta V_P}$ and $\boldsymbol{\Delta V_{PQ}}$ to $\boldsymbol{\Delta V_C}$. The matrices $\left[\frac{\partial Q_C}{\partial V_C}\right]$ and $\left[\frac{\partial Q_{PV}}{\partial V_C}\right]$ can be obtained as the corresponding submatrices of the matrix $\left[\frac{\partial Q_G}{\partial V_G}\right]$ describing the relationships between generator terminal voltages and reactive powers, which can be obtained from (5.3):

$$\left[\frac{\partial Q_G}{\partial V_G}\right] = \mathbf{B_{GG}} - \mathbf{B_{GL}}\mathbf{B_{LL}^{-1}}\mathbf{B_{LG}} \tag{C.2}$$

In (C.2), it is also assumed that $\boldsymbol{\Delta Q_L}$ is equal to zero after the disturbance activating the SVC response (the same assumption that was used to obtain (5.6) from (5.5)).

The optimization problem in (C.1) is solved at every SVC control step (e.g., every few seconds [15]) to generate a progressive terminal voltage command for the control generators. The optimal solution of (C.1) can be obtained through multi-objective constrained least squares by reformulating the objective of (C.1) as

$$\text{minimize}_{\boldsymbol{\Delta V_C}} \left\| \begin{bmatrix} \left[\frac{\partial V_P}{\partial V_C}\right] \\ \sqrt{\beta}\left(\left(\text{diag}(\overline{\boldsymbol{Q}}_{\boldsymbol{C}})\right)^{-1}\left[\frac{\partial Q_C}{\partial V_C}\right] - \left(\text{diag}(\overline{\boldsymbol{Q}}_{\boldsymbol{C}}^{\Downarrow})\right)^{-1}\left[\frac{\partial Q_C}{\partial V_C}\right]^{\Downarrow}\right) \end{bmatrix} \boldsymbol{\Delta V_C} - \begin{bmatrix} \alpha \boldsymbol{\Delta V_P} \\ \sqrt{\beta}\left(\left(\text{diag}(\overline{\boldsymbol{Q}}_{\boldsymbol{C}}^{\Downarrow})\right)^{-1}\boldsymbol{Q}_{\boldsymbol{C}}^{\Downarrow} - \left(\text{diag}(\overline{\boldsymbol{Q}}_{\boldsymbol{C}})\right)^{-1}\boldsymbol{Q_C}\right) \end{bmatrix} \right\|_2^2 \tag{C.3}$$

with the four groups of constraints from (C.1) remaining unchanged. Once the objective of (C.1) is reformulated as (C.3), standard constrained least squares solvers (e.g., the `lsqlin()` function in MATLAB) can be used to obtain the optimal trade-off solution for a given value of the trade-off parameter β [202].

Finally, the values of parameters α and β used in Chapter 5 are 0.1 and 10^{-5} respectively. Thus, the value of α is the same as in [96] and the value of β is ten times smaller than in [96] to allow for a close convergence of all pilot bus voltages to their reference values.

D

INERTIAL AGGREGATE ELECTROMECHANICAL MODEL

This appendix contains a short derivation of the reduced electromechanical model that is introduced in Section 5.3.3 and used in 5.3.5 to evaluate the accuracy of slow coherent generator groupings. The contents follow references [34, 35], which contain the full proofs and derivations of the included formulas.

The inertial and slow coherency aggregates of the model (2.19) can be obtained through transforming the original electromechanical variables in (2.19) into the *aggregate variables* of coherent areas and *area difference variables*. The aggregate variables $\boldsymbol{y}$ represent the motion of the areas' centers of inertia, and they can be obtained from the original machine rotor angles through the following transformation:

$$\begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix} = \mathbf{T^A} \left[\Delta\delta_1, \ldots, \Delta\delta_{|\mathcal{C}_1|}, \ldots, \Delta\delta_{g-|\mathcal{C}_k|+1}, \ldots, \Delta\delta_g \right]^T \tag{D.1}$$

where the transformation matrix $\mathbf{T^A}$ can be defined element-wise as:

$$\mathbf{T^A} = \begin{cases} T^A_{ij} = \frac{M_j}{\sum_{l \in \mathcal{C}_i} M_l} & \text{if } j \in \mathcal{C}_i, \\ T^A_{ij} = 0 & \text{otherwise} \end{cases} \tag{D.2}$$
$$\left(i = 1, \ldots, k; \;\; j = 1, \ldots, g\right).$$

and $\mathcal{C}_1, \ldots, \mathcal{C}_k$ are the sets of machine indices of each of the k coherent areas.

In (D.1), notice that the machine angles appear in the order of their respective areas, which is known as *sequential ordering* of the states [34, Chapter 4]. That is, the machines are re-numbered so that the machines belonging to group $\mathcal{C}_1$ are numbered 1 through $|\mathcal{C}_1|$, the machines belonging to group $\mathcal{C}_2$ are numbered $|\mathcal{C}_1|+1$ through $|\mathcal{C}_1|+|\mathcal{C}_2|$ and so on. The state ordering is crucial for the definition of the variable transformation matrices in (D.4) and (D.5), which assume the sequential ordering of machine angles.

In addition to k aggregate variables in $\boldsymbol{y}$, $g-k$ difference variables $\boldsymbol{z}$ are introduced. To define difference variables, one machine is selected in each area as a *reference machine*. With the sequential ordering of machine states as in (D.1), assume the first machine of the each area as the area's reference machine (this choice aims to simplify the expression in (D.4)). Then the $g-k$ difference variables can be expressed through the following transformation:

$$\begin{bmatrix} z_1 \\ \vdots \\ z_{g-k} \end{bmatrix} = \mathbf{T^D} \left[\Delta\delta_1, \ldots, \Delta\delta_{|\mathcal{C}_1|}, \ldots, \Delta\delta_{g-|\mathcal{C}_k|+1}, \ldots, \Delta\delta_g \right]^T \tag{D.3}$$

where the transformation matrix $\mathbf{T^D}$ is given by:

$$\mathbf{T^D} = \text{blkdiag}\left(\left[-\mathbf{1}_{|\mathcal{C}_1|-1} \;\; \mathbf{I}_{|\mathcal{C}_1|-1} \right], \left[-\mathbf{1}_{|\mathcal{C}_2|-1} \;\; \mathbf{I}_{|\mathcal{C}_2|-1} \right], \ldots, \left[-\mathbf{1}_{|\mathcal{C}_k|-1} \;\; \mathbf{I}_{|\mathcal{C}_k|-1} \right] \right) \tag{D.4}$$

If a machine group $\mathcal{C}$ consists of a single machine (i.e., $|\mathcal{C}|-1=0$), there will be no difference variables corresponding to this group, and the column in the matrix $\mathbf{T^D}$ corresponding to this group will be zero.

From (D.1) and (D.3), the complete variable transformation is given as:

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{bmatrix} = \begin{bmatrix} \mathbf{T^A} \\ \mathbf{T^D} \end{bmatrix} \boldsymbol{\Delta\delta} = \mathbf{T}\boldsymbol{\Delta\delta} \tag{D.5}$$

By applying the transformation $\mathbf{T}$ to (2.19), the following system emerges:

$$\begin{bmatrix} \ddot{\boldsymbol{y}} \\ \ddot{\boldsymbol{z}} \end{bmatrix} = \left[\begin{array}{c|c} \mathbf{F_{11}} & \mathbf{F_{12}} \\ \hline \mathbf{F_{21}} & \mathbf{F_{22}} \end{array} \right] \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{bmatrix} \tag{D.6}$$

where, based on (D.5) and (5.10), $\mathbf{F}$ is given as:

$$\mathbf{F} = \mathbf{T}\mathbf{M}^{-1}\mathbf{K}\mathbf{T}^{-1} \tag{D.7}$$

If the coherency grouping $\mathcal{C}_1, \ldots, \mathcal{C}_k$ underlying the transformation (D.5) is ideal, the off-diagonal matrices $\mathbf{F_{12}}$ and $\mathbf{F_{21}}$ will be zero, and the spectrum of the original model (2.19) will be perfectly captured by the subsystem $\mathbf{F_{11}}$ representing the aggregate dynamics of the k machine groups and the subsystem $\mathbf{F_{22}}$ representing the local dynamics inside the groups. However, for the practical power systems, the perfect decoupling between the aggregate area variables $\boldsymbol{y}$ and the local difference variables $\boldsymbol{z}$ is not achievable (i.e., $\mathbf{F_{12}}$ and $\mathbf{F_{21}}$ are not zero in practice). Therefore, in the case of slow coherency grouping, the spectrum of $\mathbf{F_{11}}$ will not perfectly match the k slowest eigenvalues of $\mathbf{M}^{-1}\mathbf{K}$, but the eigenvalue difference may be smaller or larger depending on the actual machine grouping underlying the transformation (D.5). Furthermore, the following model

$$\ddot{\boldsymbol{y}} = \mathbf{F_{11}}\boldsymbol{y} \tag{D.8}$$

is known in the literature [34, 35] as the *inertial aggregate model*. It has a clear physical meaning, as it can be obtained from the original electromechanical model by linking the internal nodes of machines in the same group by infinite admittances.

NOMENCLATURE

ACRONYMS

AC	alternating current
ACE	area control error
AGC	automatic generation control
AHC	agglomerative hierarchical clustering
AVR	automatic voltage regulator
AZD	adaptive zone division
BFS	breadth-first search
CLI	command-line interface
CSVC	coordinated secondary voltage control
DC	direct current
DSA	dynamic security assessment
DSP	digital signal processing
EHV	extra high voltage
ERPI	Electric Power Research Institute
FACTS	flexible AC transmission systems
FCSC	flexible constrained spectral clustering [191]
FLDF	fast decoupled load flow
GD	gradient descent
HSC	hierarchical spectral clustering
HV	high voltage
HVDC	high voltage DC
ICI	intentional controlled islanding
ICT	information and communications technology
LFC	load-frequency control
LP	linear programming
MAE	mean absolute error
MILP	mixed-integer linear programming
MINLP	mixed-integer nonlinear programming
MIP	mixed-integer programming
MST	minimum spanning tree
MVDC	medium voltage DC
NADIR	lowest system frequency value following a disturbance
NJW	spectral clustering algorithm of Ng, Jordan, and Weiss
NPCC	Northeast Power Coordinating Council
NSCOGI	North Seas Countries Offshore Grid Initiative
OLTC	on load tap changer

OPF	optimal power flow
PDC	phasor data concentrator
PF	power flow
PMU	phasor measurement unit
PPSR	parallel power system restoration
PST	Power System Toolbox
PV	photovoltaics
PVC	primary voltage control
RA	remedial action
RAS	remedial action scheme
RMS	root mean square
RMSE	root mean square error
ROCOF	rate of change of frequency
SCADA	supervisory control and data acquisition
SIPS	system integrity protection scheme
SMT	synchronized measurement technology
SpMST	spectral minimum spanning tree
SPS	special protection system
SVC	secondary voltage control
SVD	singular value decomposition
TEP	transmission expansion planning
TSO	transmission system operator
TVC	tertiary voltage control
UC	unit commitment
VCA	voltage control area
VCS	Var control space
VCZ	voltage control zone
VSA	voltage security assessment
V-WAP	wide-area voltage protection
WAMC	wide-area monitoring and control
WAMPAC	wide-area monitoring, protection and control
WAMS	wide-area monitoring system
WE	wind energy

VECTORS AND MATRICES

$\mathbf{C}$	graph incidence matrix
$\mathbf{L}$	graph Laplacian
$\mathbf{L_{rw}}$	random walk normalized Laplacian
$\mathbf{L_n}$	symmetric normalized Laplacian
$\mathbf{P}$	random walk transition matrix
$\mathbf{R}$	orthogonal transformation matrix
$\mathbf{W}$	weighted graph adjacency matrix
$\mathbf{W_n}$	normalized weighted graph adjacency matrix
$\mathbf{I}$	identity matrix

$\mathbf{1}$	all-ones matrix
$\mathbf{0}$	all-zeros matrix
$\mathbb{1}$	cluster indicator vector
$\mathbf{Q}$	constraint matrix in FCSC
$\mathbf{Q_n}$	normalized constraint matrix in FCSC
$\mathbf{K}$	matrix of synchronizing torque coefficients
$\mathbf{K^S}$	symmetrized matrix of synchronizing torque coefficients
$\mathbf{K^B}$	symmetric matrix of synchronizing torque coefficients obtained by neglecting conductances in the Kron-reduced admittance matrix

SETS

$\mathcal{C}$	set of objects belonging to the same cluster or group
$\mathcal{E}$	set of graph edges
$\mathcal{G}$	set of machine nodes
$\mathcal{L}$	set of electric power load nodes
$\mathcal{T}$	set of terminal nodes forming a node grouping constraint
$\mathcal{V}$	set of all graph nodes

OTHER SYMBOLS

D	weighted graph node degree
$\mathscr{G}$	graph (mathematical structure)
g	number of synchronous machines
k	number of clusters
m	number of graph edges
n	number of graph nodes
$\mathscr{T}$	tree graph (mathematical structure)
α	constraint threshold in FCSC; GD learning rate
β	lower column threshold in clustering algorithms based on aligned spectral embeddings
ε	minimal cluster size as percentage of the average cluster size
η	number of misplaced terminal nodes
μ	generic graph node weight
ϕ	cluster expansion ratio
ϕ_{max}	maximal cluster expansion ratio over all clusters

PHYSICAL PARAMETERS

B_{ij}	susceptance between nodes v_i and v_j
G_{ij}	conductance between nodes v_i and v_j
H_i	synchronous machine inertia of machine at node v_i
K_{ij}	synchronizing torque coefficient between synchronous machines i and j

$P_{G,i}$	generated active power output at node v_i
$P_{i,j}^{max}$	maximal active power flow limit between nodes v_i and v_j
P_m	synchronous machine mechanical power
$Q_{G,i}$	generated reactive power output at node v_i
V_f	synchronous machine excitation voltage
V_G	generator terminal voltage
V_P	pilot bus voltage
x'_d	synchronous machine transient reactance
ξ	synchronous machine damping coefficient
δ	synchronous machine rotor angle or bus voltage angle
ω_0	nominal power system frequency
ω	synchronous machine rotor speed
M_i	synchronous machine scaled inertia of machine at node v_i

OPTIMIZATION VARIABLES

$P_{GS,i}$	generation adjustment (increase or decrease) at node v_i
$P_{i,j}$	active power flow between nodes v_i and v_j
$P_{LS,i}$	load shedding amount at node v_i
$T_{i,j}$	artificial commodity flow between nodes v_i and v_j
$x_{i,l}$	binary variable indicating the membership of node v_i in cluster l
$y_{i,j}$	binary variable indicating the open or closed status of branch (v_i, v_j)
$z_{i,j,l}$	binary variable indicating whether both nodes v_i and v_j belong to cluster l

INDICES

$*$	Indicates an optimal value
ref	Indicates a reference value

BIBLIOGRAPHY

[1] P. Kundur. *Power system stability and control*. 1st edition. McGraw-Hill, 1994.

[2] E. F. Camacho, T. Samad, M. Garcia-Sanz, and I. Hiskens. "Control for Renewable Energy and Smart Grids". In: *The Impact of Control Technology*. Ed. by T. Samad and A. M. Annaswamy. IEEE Control Systems Society, 2011, pp. 79–98.

[3] V. Terzija, G. Valverde, D. Cai, P. Regulski, V. Madani, J. Fitch, et al. "Wide-area monitoring, protection, and control of future electric power networks". In: *Proc. IEEE* 99.1 (2011), pp. 80–93.

[4] P. Ceferin, M. Naglič, and A. Souvent. "Wide Area Monitoring Systems and Information and Communication Technology Networks (WAMS systems and ICT networks)". In: *CIGRE Science and Engineering* 6.5 (2016), pp. 109–118.

[5] J. Arrillaga, M. H. J. Bollen, and N. R. Watson. "Power quality following deregulation". In: *Proc. IEEE* 88.2 (2000), pp. 246–261.

[6] D. Kirschen and G. Strbac. "Why Investments Do Not Prevent Blackouts". In: *The Electricity Journal* 17.2 (2004), pp. 29–36.

[7] European Commission. *Vision and Strategy for Europe's Electricity Networks of the Future*. 2006. URL: `http://ec.europa.eu`. (last accessed: 31-10-2018).

[8] M. A. M. M. van der Meijden. *A sustainable and reliable electricity system; Inevitable and challenging*. Tech. rep. Department of EEMCS, Delft University of Technology, 2012, pp. 1–34.

[9] UCTE. *FINAL REPORT of the Investigation Committee on the 28 September 2003 Blackout in Italy*. Tech. rep. Union for the Coordination of the Transmission of Electricity (UCTE), 2004, pp. 1–128.

[10] G. Kendall. "Power outages during market deregulation". In: *IEEE Control Syst. Mag.* 21.6 (2001), pp. 33–39.

[11] H. Pandžić, Y. Dvorkin, T. Qiu, Y. Wang, and D. S. Kirschen. "Toward Cost-Efficient and Reliable Unit Commitment Under Uncertainty". In: *IEEE Trans. Power Syst.* 31.2 (2015), pp. 970–982.

[12] A. R. Ciupuliga. "Transmission expansion planning under increased uncertainties; towards efficient and sustainable power systems". PhD Thesis. Delft University of Technology, Faculty of EEMCS, 2013.

[13] S. S. Torbaghan, M. Gibescu, B. G. Rawn, and M. A. M. M. van der Meijden. "A Market-Based Transmission Planning for HVDC Grid — Case Study of the North Sea". In: *IEEE Trans. on Power Syst.* 30.2 (2015), pp. 784–794.

[14] M. O. Buygi, G. Balzer, H. M. Shanechi, and M. Shahidehpour. "Market-based Transmission Expansion Planning". In: *IEEE Trans. Power Syst.* 19.4 (2004), pp. 2060–2067.

[15] H. Vu, P. Pruvot, C. Launay, and Y. Harmand. "An Improved Voltage Control on Large-Scale Power System". In: *IEEE Trans. Power Syst.* 11.3 (1996), pp. 1295–1303.

[16] M. McDonald and D. Tziouvaras. *Power swing and out-of-step considerations on transmission lines; A report to the Power System Relaying Committee Of the IEEE Power Engineering Society*. Tech. rep. IEEE PSRC WG D6, 2005, pp. 1–59.

[17] IEEE Power & Energy Society. *Blackout Experiences and Lessons, Best Practices for System Dynamic Performance, and the Role of New Technologies*. IEEE Task Force Report. IEEE, 2007, pp. 1–233.

[18] Transelectrica & Politecnico di Torino. *Analysis of historic outages*. SESAME project report D1.1. Version 2.0. 2011, pp. 1–107.

[19] S. Norris. "Preventing Wide Area Blackouts in Transmission Systems: A New Approach for Intentional Controlled Islanding using Power Flow Tracing". PhD Thesis. Durham University, School of Engineering and Computing Sciences, 2014.

[20] O. P. Veloza and F. Santamaria. "Analysis of Major Blackouts from 2003 to 2015: Classification of Incidents and Review of Main Causes". In: *The Electricity Journal* 29 (2016), pp. 42–49.

[21] Wikipedia. *List of major power outages*. URL: `http://en.wikipedia.org/wiki/List_of_major_power_outages`. (last accessed: 08-11-2018).

[22] Wikipedia. *Power outages in Malaysia*. URL: `http://en.wikipedia.org/wiki/Power_outages_in_Malaysia`. (last accessed: 09-11-2018).

[23] G. Andersson, P. Donalek, R. Farmer, N. Hatziargyriou, I. Kamwa, P. Kundur, et al. "Causes of the 2003 Major Grid Blackouts in North America and Europe, and Recommended Means to Improve System Dynamic Performance". In: *IEEE Trans. Power Syst.* 20.4 (2005), pp. 1922–1928.

[24] Y. V. Makarov, V. I. Reshetov, V. A. Stroev, and N. I. Voropai. "Blackouts in North America and Europe: Analysis and Generalization". In: *Proc. 6th IEEE PES PowerTech Conference (PowerTech 2005)*. St. Petersburg, Russia, 2005, pp. 1–7.

[25] A. G. Phadke and J. S. Thorp. *Synchronized Phasor Measurements and Their Applications*. Springer US, 2008.

[26] IEEE Power & Energy Society. *IEEE Std C37.118.1-2011 (Revision of IEEE Std C37.118-2005)*. IEEE Standard for Synchrophasor Measurements for Power Systems. IEEE, 2011, pp. 1–61.

[27] IEEE Power Engineering Society. *IEEE Std C37.1-2007 (Revision of IEEE Std C37.1-1994)*. IEEE Standard for SCADA and Automation Systems. IEEE, 2008, pp. 1–146.

[28] S. N. D. R. Nuthalapati, ed. *Power System Grid Operation Using Synchrophasor Technology*. Springer International Publishing, 2019.

[29] D. Kosterev. "Synchrophasor Technology at BPA; From Wide-Area Monitoring to Wide-Area Control". In: *Power System Grid Operation Using Synchrophasor Technology*. Ed. by S. N. D. R. Nuthalapati. Springer International Publishing, 2019, pp. 77–127.

[30] E. Cotilla-Sanchez, P. D. H. Hines, C. Barrows, S. Blumsack, and M. Patel. "Multi-Attribute Partitioning of Power Networks Based on Electrical Distance". In: *IEEE Trans. Power Syst.* 28.4 (2013), pp. 4979–4987.

[31] S. Blumsack, P. D. H. Hines, M. Patel, C. Barrows, and E. Cotilla-Sanchez. "Defining power network zones from measures of electrical distance". In: *Proc. Power & Energy Society General Meeting (PES GM 2009)*. 2009, pp. 1–8.

[32] S. B. Yusof, G. J. Rogers, and R. T. H. Alden. "Slow coherency based network partitioning including load buses". In: *IEEE Trans. Power Syst.* 8.3 (1993), pp. 1375–1382.

[33] B. Avramovic, P. V. Kokotovic, J. R. Winkelman, and J. H. Chow. "Area decomposition for electromechanical models of power systems". In: *Automatica* 16.6 (1980), pp. 637–648.

[34] J. H. Chow. *Time-scale modeling of dynamic networks with applications to power systems*. Vol. 46. Lecture notes in control and information sciences. Berlin and New York: Springer-Verlag, 1982.

[35] J. H. Chow, ed. *Power System Coherency and Model Reduction*. Springer-Verlag New York, 2013, pp. 1–308.

[36] K. Morison, X. Wang, A. Moshref, and A. Edris. "Identification of voltage control areas and reactive power reserve; An advancement in online voltage security assessment". In: *Proc. Power & Energy Society General Meeting (PES GM 2008)*. Pittsburgh, Pennsylvania, USA, 2008, pp. 1–7.

[37] V. Quintana and N. Müller. "Partitioning of power networks and applications to security control". In: *Proc. Inst. Elect. Eng. C* 138.6 (1991), pp. 535–545.

[38] K. Morison and L. Wang. "Reduction of Large Power System Models: A Case Study". In: *Power System Coherency and Model Reduction*. Ed. by J. H. Chow. Springer-Verlag New York, 2013, pp. 143–158.

[39] I. Kamwa, A. K. Pradhan, G. Joos, and S. R. Samantaray. "Fuzzy partitioning of a real power system for dynamic vulnerability assessment". In: *IEEE Trans. Power Syst.* 24.3 (2009), pp. 1356–1365.

[40] K. Mei, S. M. Rovnyak, and C.-M. Ong. "Clustering-based dynamic event location using wide-area phasor measurements". In: *IEEE Trans. Power Syst.* 23.2 (2008), pp. 673–679.

[41] S. Corsi. "Wide area voltage protection". In: *IET Generation, Transmission & Distribution* 4.10 (2010), pp. 1164–1179.

[42] H. Li, G. W. Rosenwald, J. Jung, and C.-C. Liu. "Strategic Power Infrastructure Defense". In: *Proc. IEEE* 93.5 (2005), pp. 918–933.

[43] L. Ding, F. M. Gonzalez-Longatt, P. Wall, and V. Terzija. "Two-Step Spectral Clustering Controlled Islanding Algorithm". In: *IEEE Trans. Power Syst.* 28.1 (2013), pp. 75–84.

[44] M. M. Adibi and L. H. Fink. “Power System Restoration Planning”. In: *IEEE Trans. Power Syst.* 9.1 (1994), pp. 22–28.

[45] J. Quirós-Tortós, M. Panteli, P. Wall, and V. Terzija. “Sectionalising methodology for parallel system restoration based on graph theory”. In: *IET Generation, Transmission & Distribution* 9.11 (2015), pp. 1216–1225.

[46] P. Lagonotte, J. Sabonnadiere, J.-Y. Leost, and J.-P. Paul. “Structural analysis of the electrical system: Application to secondary voltage control in France”. In: *IEEE Trans. Power Syst.* 4.2 (1989), pp. 479–486.

[47] S. Corsi, M. Pozzi, C. Sabelli, and A. Serrani. “The coordinated automatic voltage control of the Italian transmission grid – Part I: Reasons of the choice and overview of the consolidated hierarchical system”. In: *IEEE Trans. Power Syst.* 19.4 (2004), pp. 1723–1732.

[48] S. Corsi. “Wide Area Voltage Protection”. In: *Voltage Control and Protection in Electrical Power Systems*. Springer-Verlag London, 2015, pp. 497–542.

[49] T. Kyriakidis, R. Cherkaoui, and M. Kayal. “Generator coherency identification algorithm using modal and time-domain information”. In: *Proc. 15th IEEE International Conference on Smart Technologies (EUROCON 2013)*. Zagreb, Croatia, 2013, pp. 1–8.

[50] M. A. M. Ariff and B. C. Pal. “Coherency Identification in Interconnected Power System – An Independent Component Analysis Approach”. In: *IEEE Trans. Power Syst.* 28.2 (2013), pp. 1747–1755.

[51] S. Corsi. *Voltage Control and Protection in Electrical Power Systems*. Springer-Verlag London, 2015.

[52] H. Sun, Q. Guo, B. Zhang, W. Wu, and B. Wang. “An adaptive zone-division-based automatic voltage control system with applications in China”. In: *IEEE Trans. Power Syst.* 28.2 (2013), pp. 1816–1828.

[53] L. Ding, Z. Ma, P. Wall, and V. Terzija. “Graph spectra based controlled islanding for low inertia power systems”. In: *IEEE Trans. Power Del.* 32.1 (2017), pp. 302–309.

[54] J. Quirós-Tortós, P. Wall, L. Ding, and V. Terzija. “Determination of sectionalising strategies for parallel power system restoration: A spectral clustering-based methodology”. In: *Electric Power Systems Research* 116 (2014), pp. 381–390.

[55] H. You, V. Vittal, and X. Wang. “Slow Coherency-Based Islanding”. In: *IEEE Trans. Power Syst.* 19.1 (2004), pp. 483–491.

[56] B. Yang, V. Vittal, and G. T. Heydt. “Slow-Coherency-Based Controlled Islanding—A Demonstration of the Approach on the August 14, 2003 Blackout Scenario”. In: *IEEE Trans. Power Syst.* 21.4 (2006), pp. 1840–1847.

[57] C. G. Wang, B. H. Zhang, Z. G. Hao, J. Shu, P. Li, and Z. Q. Bo. “A Novel Real-Time Searching Method for Power System Splitting Boundary”. In: *IEEE Trans. Power Syst.* 25.4 (2010), pp. 1902–1909.

[58] G. Xu and V. Vittal. “Slow coherency based cutset determination algorithm for large power systems”. In: *IEEE Trans. Power Syst.* 25.2 (2010), pp. 877–884.

[59] J. Quirós-Tortós, R. Sánchez-García, J. Brodzki, J. Bialek, and V. Terzija. “Constrained spectral clustering-based methodology for intentional controlled islanding of large-scale power systems”. In: *IET Generation, Transmission & Distribution* 9.1 (2015), pp. 31–42.

[60] A. Esmaeilian and M. Kezunovic. “Controlled islanding to prevent cascade outages using constrained spectral k-embedded clustering”. In: *Proc. 19th Power Systems Computation Conference (PSCC 2016)*. Genova, Italy, 2016, pp. 1–6.

[61] A. Esmaeilian and M. Kezunovic. “Prevention of Power Grid Blackouts Using Intentional Islanding Scheme”. In: *IEEE Trans. Ind. Appl.* 53.1 (2017), pp. 622–629.

[62] G. Morales-España. “Unit Commitment: Computational Performance, System Representation and Wind Uncertainty Management”. PhD Thesis. Pontifical Comillas University, KTH Royal Institute of Technology, and Delft University of Technology, 2014.

[63] S. Lumbreras. “Decision support methods for large-scale flexible transmission expansion planning”. PhD Thesis. Pontifical Comillas University, 2014.

[64] Y. Dvorkin. “Operations and Planning in Sustainable Power Systems”. PhD Thesis. University of Washington, 2016.

[65] K. Andreev and H. Räcke. “Balanced graph partitioning”. In: *Proc. 16th annual ACM symposium on Parallelism in algorithms and architectures (SPAA 2004)*. Barcelona, Spain, 2004, pp. 120–124.

[66] C. Schulz. *Graph Partitioning and Graph Clustering in Theory and Practice*. Course Notes. Institute for Theoretical Informatics, Karlsruhe Institute of Technology (KIT), 2016, pp. 1–202.

[67] C. Hamon, E. Shayesteh, M. Amelin, and L. Söder. "Two partitioning methods for multi-area studies in large power systems". In: *International Transactions on Electrical Energy Systems* 25.4 (2015), pp. 648–660.

[68] S. M. van Dongen. "Graph Clustering by Flow Simulation". PhD Thesis. Utrecht University, Faculteit Wiskunde en Informatica, 2000.

[69] T. De Bie, J. Suykens, and B. De Moor. "Learning from General Label Constraints". In: *Structural, Syntactic, and Statistical Pattern Recognition. SSPR/SPR 2004*. Ed. by A. Fred, T. M. Caelli, R. P. W. Duin, A. C. Campilho, and D. De Ridder. Vol. 3138. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2004, pp. 671–679.

[70] U. von Luxburg. "Statistical Learning with Similarity and Dissimilarity Functions". PhD Thesis. Technical University of Berlin, Faculty of Electrical Engineering and Informatics, 2004.

[71] T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: Data mining, inference, and prediction*. 2nd edition, corrected 7th printing. Springer series in statistics. New York: Springer, 2009.

[72] J. A. S. Almeida, L. M. S. Barbosa, A. A. C. C. Pais, and S. J. Formosinho. "Improving hierarchical cluster analysis: A new method with outlier detection and automatic clustering". In: *Chemometrics and Intelligent Laboratory Systems* 87.2 (2007), pp. 208–217.

[73] F. R. K. Chung, ed. *Spectral graph theory*. Number 92 in CBMS Regional Conference Series in Mathematics. American Mathematical Society, 1997.

[74] U. von Luxburg. "A tutorial on spectral clustering". In: *Statist. Comput.* 17.4 (2007), pp. 395–416.

[75] D. Aloise, A. Deshpande, P. Hansen, and P. Popat. "NP-hardness of Euclidean sum-of-squares clustering". In: *Machine Learning* 75.2 (2009), pp. 245–248.

[76] R. J. Sanchez-Garcia, M. Fennelly, S. Norris, N. Wright, G. Niblo, J. Brodzki, et al. "Hierarchical Spectral Clustering of Power Grids". In: *IEEE Trans. Power Syst.* 29.5 (2014), pp. 2229–2237.

[77] M. Grötschel, A. Martin, and R. Weismantel. "The Steiner Tree packing problem in VLSI design". In: *Mathematical Programming* 78.2 (1997), pp. 265–281.

[78] M. Grötschel, A. Martin, and R. Weismantel. "Packing Steiner Trees: Separation Algorithms". In: *SIAM Journal on Discrete Mathematics* 9.2 (1996), pp. 233–257.

[79] A. Y. Ng, M. I. Jordan, and Y. Weiss. "On spectral clustering: Analysis and an algorithm". In: *Proc. Advances in Neural Information Processing Systems 14 (NIPS 2001)*. 2001, pp. 849–856.

[80] S. X. Yu and J. Shi. "Grouping with Bias". In: *Proc. Advances in Neural Information Processing Systems 14 (NIPS 2001)*. Ed. by T. G. Dietterich, S. Becker, and Z. Ghahramani. 2002, pp. 1327–1334.

[81] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas. "MATPOWER: Steady-State Operations, Planning, and Analysis Tools for Power Systems Research and Education". In: *IEEE Trans. Power Syst.* 26.1 (2011), pp. 12–19.

[82] MATLAB. *version 9.2.0 (R2017a)*. Available at `http://www.mathworks.com` (Online). Natick, Massachusetts: The MathWorks Inc., 2017.

[83] J. H. Chow and K. W. Cheung. "A toolbox for power system dynamics and control engineering education and research". In: *IEEE Trans. Power Syst.* 7.4 (1992), pp. 1559–1564.

[84] DIgSILENT GmbH. *PowerFactory 2018 User Manual*. Tech. rep. Online Edition, available at `http://www.digsilent.de`. Gomaringen, Germany, 2018.

[85] G. van Rossum. *Python tutorial*. Tech. rep. CS-R9526. Amsterdam: Centrum voor Wiskunde en Informatica (CWI), May 1995.

[86] P. M. Anderson and A. A. Fouad. *Power system control and stability*. 2nd ed. IEEE Press power engineering series. Piscataway, N.J.: IEEE Press and Wiley-Interscience, 2003.

[87] R. E. Rosenthal. *GAMS – A Users Guide*. Washington, DC, USA: GAMS Development Corporation, 2016.

[88] M. Bastian, S. Heymann, and M. Jacomy. "Gephi: An Open Source Software for Exploring and Manipulating Networks". In: *Proc. 3rd International AAAI Conference on Weblogs and Social Media (AAAI ICWSM 2009)*. San Jose, California, USA, 2009, pp. 1–2.

[89] J. Ellson, E. Gansner, L. Koutsofios, S. C. North, and G. Woodhull. “Graphviz – Open Source Graph Drawing Tools”. In: *Graph Drawing. GD 2001*. Ed. by P. Mutzel, M. Jünger, and S. Leipert. Vol. 2265. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2002, pp. 483–484.

[90] G. Karypis and V. Kumar. “A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs”. In: *SIAM Journal on Scientific Computing* 20.1 (1998), pp. 359–392.

[91] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar. “Multilevel hypergraph partitioning: applications in VLSI domain”. In: *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* 7.1 (1999), pp. 69–79.

[92] P. Sanders and C. Schulz. “Think Locally, Act Globally: Highly Balanced Graph Partitioning”. In: *Proc. 12th International Symposium on Experimental Algorithms (SEA 2013)*. Vol. 7933. LNCS. Springer, 2013, pp. 164–175.

[93] S. Corsi, F. De Villiers, and R. Vajeth. “Secondary Voltage Regulation applied to the South Africa transmission grid”. In: *Proc. Power & Energy Society General Meeting (PES GM 2010)*. Providence, RI, USA, 2010, pp. 1–8.

[94] V. Alimisi and P. C. Taylor. “Zoning evaluation for improved coordinated automatic voltage control”. In: *IEEE Trans. Power Syst.* 30.5 (2015), pp. 2736–2746.

[95] A. Conejo, T. Gomez, and J. de la Fuente. “Pilot-bus selection for secondary voltage control”. In: *Eur. Trans. Electr. Power Eng.* 3.5 (1993), pp. 359–366.

[96] A. Conejo and M. Aguilar. “Secondary voltage control: Nonlinear selection of pilot buses, design of an optimal control law, and simulation results”. In: *IEE Proc. Gen., Transm., Distrib.* 145.1 (1998), pp. 77–81.

[97] G. Karypis, E.-H. Han, and V. Kumar. “Chameleon: Hierarchical clustering using dynamic modeling”. In: *Computer* 32.8 (1999), pp. 68–75.

[98] E. Hartuv and R. Shamir. “A clustering algorithm based on graph connectivity”. In: *Information Processing Letters* 76.4–6 (2000), pp. 175–181.

[99] C. T. Zahn. “Graph-theoretical methods for detecting and describing gestalt clusters”. In: *IEEE Trans. Comput.* 100.1 (1971), pp. 68–86.

[100] V. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. “Fast unfolding of communities in large networks”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2008.10 (2008), pp. 1–12.

[101] M. Girvan and M. E. J. Newman. “Community structure in social and biological networks”. In: *Proc. Natl. Acad. Sci. U S A.* 99.12 (2002), pp. 7821–7826.

[102] G. W. Flake, R. E. Tarjan, and K. Tsioutsiouliklis. “Graph clustering and minimum cut trees”. In: *Internet Mathematics* 1.4 (2004), pp. 385–408.

[103] S. E. Schaeffer. “Graph clustering”. In: *Computer Science Review* 1.1 (2007), pp. 27–64.

[104] J. Leskovec, K. J. Lang, and M. Mahoney. “Empirical comparison of algorithms for network community detection”. In: *Proc. of the 19th International Conference on World Wide Web (WWW 2010)*. Raleigh, North Carolina, USA, 2010, pp. 1–11.

[105] M. Fiedler. “A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory”. In: *Czechoslovak Mathematical Journal* 25.4 (1975), pp. 619–633.

[106] C. M. Fiduccia and R. M. Mattheyses. “A Linear-Time Heuristic for Improving Network Partitions”. In: *Proc. 19th Design Automation Conference (DAC 1982)*. Las Vegas, Nevada, USA, 1982, pp. 1–7.

[107] J. Li, C.-C. Liu, and K. P. Schneider. “Controlled Partitioning of a Power Network Considering Real and Reactive Power Balance”. In: *IEEE Trans. Smart Grid* 1.3 (2010), pp. 261–269.

[108] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. Wiley, 1997.

[109] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. “The complexity of multiterminal cuts”. In: *SIAM Journal on Computing* 23.4 (1994), pp. 864–894.

[110] K. Lang and S. Rao. “A flow-based method for improving the expansion or conductance of graph cuts.” In: *Proc. of the 10th International Conference on Integer Programming and Combinatorial Optimization (IPCO 2004)*. Ed. by D. Bienstock and G. Nemhauser. Vol. 3064. LNCS. Springer, 2004, pp. 325–337.

[111] B. W. Kernighan and S. Lin. “An efficient heuristic procedure for partitioning graphs”. In: *The Bell System Technical Journal* 49.2 (1970), pp. 291–307.

[112] N. Fan, D. Izraelevitz, F. Pan, P. M. Pardalos, and J. Wang. “A mixed integer programming approach for optimal power grid intentional islanding”. In: *Energy Systems* 3.1 (2012), pp. 77–93.

[113] P. A. Trodden, W. A. Bukhsh, A. Grothey, and K. I. M. McKinnon. “MILP formulation for controlled islanding of power networks”. In: *International Journal of Electrical Power & Energy Systems* 45.1 (2013), pp. 501–508.

[114] W. Sun, C.-C. Liu, and L. Zhang. “Optimal Generator Start-Up Strategy for Bulk Power System Restoration”. In: *IEEE Trans. Power Syst.* 26.3 (2011), pp. 1357–1366.

[115] C. S. Chang, L. R. Lu, and F. S. Wen. “Power system network partitioning using tabu search”. In: *Electric Power Systems Research* 49.1 (1999), pp. 55–61.

[116] M. R. Aghamohammadi and A. Shahmohammadi. “Intentional islanding using a new algorithm based on ant search mechanism”. In: *International Journal of Electrical Power & Energy Systems* 35.1 (2012), pp. 138–147.

[117] M. Paramasivam, S. Dasgupta, V. Ajjarapu, and U. Vaidya. “Contingency Analysis and Identification of Dynamic Voltage Control Areas”. In: *IEEE Trans. Power Syst.* 30.6 (2015), pp. 2974–2983.

[118] D. Vercamer, B. Steurtewagen, D. van den Poel, and F. Vermeulen. “Predicting Consumer Load Profiles Using Commercial and Open Data”. In: *IEEE Trans. Power Syst.* 31.5 (2016), pp. 3693–3701.

[119] S. Lin, F. Li, E. Tian, Y. Fu, and D. Li. “Clustering Load Profiles for Demand Response Applications”. In: *IEEE Trans. Smart Grid* Early Access (2018), pp. 1–9.

[120] J. Guo, G. Hug, and O. K. Tonguz. “Intelligent Partitioning in Distributed Optimization of Electric Power Systems”. In: *IEEE Trans. Smart Grid* 7.3 (2016), pp. 1249–1258.

[121] J. H. Chow. “New algorithms for slow coherency aggregation of large power systems”. In: *Systems and Control Theory for Power Systems*. Ed. by J. H. Chow, P. V. Kokotović, and R. J. Thomas. Vol. 64. IMA volumes in mathematics and its applications. New York: Springer-Verlag, 1995.

[122] S. Yu and J. Shi. “Multiclass spectral clustering”. In: *Proc. 9th International Conference on Computer Vision (ICCV 2003)*. 2003, pp. 313–319.

[123] J. Gallier. *Spectral Theory of Unsigned and Signed Graphs. Applications to Graph Clustering: a Survey*. Lecture Notes. Department of Computer and Information Science, University of Pennsylvania, 2015, pp. 1–122.

[124] I. S. Dhillon, Y. Guan, and B. Kulis. “Weighted graph cuts without eigenvectors: a multilevel approach”. In: *IEEE Trans. Pattern Anal. Mach. Intell* 29.11 (2007).

[125] M. Stoer and F. Wagner. “A simple min-cut algorithm”. In: *JACM* 44.4 (1997), pp. 585–591.

[126] S. S. Rangapuram, P. K. Mudrakarta, and M. Hein. “Tight Continuous Relaxation of the Balanced k-Cut Problem”. In: *Proc. Advances in Neural Information Processing Systems 27 (NIPS 2014)*. 2014, pp. 3131–3139.

[127] J. Shi and J. Malik. “Normalized cuts and image segmentation”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 22.8 (2000), pp. 888–905.

[128] G. H. Golub and C. F. Van Loan. *Matrix computations*. Vol. 3. JHU Press, 2012.

[129] L. Zelnik-Manor and P. Perona. “Self-Tuning Spectral Clustering”. In: *Proc. Advances in Neural Information Processing Systems 17 (NIPS 2004)*. 2004, pp. 1601–1608.

[130] A. Bondy and U. S. R. Murty. *Graph Theory*. Vol. 244. Graduate Texts in Mathematics. London: Springer, 2011. 655 pp.

[131] J. Machowski, J. Bialek, and J. Bumby. *Power System Dynamics: Stability and Control*. 2nd ed. Wiley, 2008. 658 pp.

[132] N. Senroy. “Generator Coherency Using the Hilbert–Huang Transform”. In: *IEEE Trans. Power Syst.* 23.4 (2008), pp. 1701–1708.

[133] X. Wang, V. Vittal, and G. T. Heydt. “Tracing generator coherency indices using the continuation method: a novel approach”. In: *IEEE Trans. Power Syst.* 20.3 (2005), pp. 1510–1518.

[134] G. Rogers. *Power System Oscillations*. Power Electronics and Power Systems. Springer US, 2000. XI, 328.

[135] H. Song, J. Wu, and K. Wu. “A wide-area measurement systems-based adaptive strategy for controlled islanding in bulk power systems”. In: *Energies* 7 (4 2014), pp. 2631–2657.

[136] H. K. Temraz, M. M. A. Salama, and V. H. Quintana. “Application of partitioning techniques for decomposing large-scale electric power networks”. In: *International Journal of Electrical Power & Energy Systems* 16 (5 1994), pp. 301–309.

[137] P. Demetriou, J. Quirós-Tortós, E. Kyriakides, and V. Terzija. “On implementing a spectral clustering controlled islanding algorithm in real power systems”. In: *Proc. 9th IEEE PES PowerTech Conference (PowerTech 2013)*. Grenoble, France, 2013, pp. 1–6.

[138] H. Meyerhenke, P. Sanders, and C. Schulz. “Partitioning complex networks via size-constrained clustering”. In: *Proc. of the 13th International Symposium on Experimental Algorithms (SEA 2014)*. Ed. by J. Gudmundsson and J. Katajainen. Vol. 8504. LNCS. Springer, 2014, pp. 351–363.

[139] A. Kyriacou, P. Demetriou, C. Panayiotou, and E. Kyriakides. “Controlled Islanding Solution for Large-Scale Power Systems”. In: *IEEE Trans. Power Syst.* 33.2 (2018), pp. 1591–1602.

[140] S. Fliscounakis, P. Panciatici, F. Capitanescu, and L. Wehenkel. “Contingency Ranking With Respect to Overloads in Very Large Power Systems Taking Into Account Uncertainty, Preventive, and Corrective Actions”. In: *IEEE Trans. Power Syst.* 28.4 (2013), pp. 4909–4917.

[141] B. Pal and B. Chaudhuri. *Robust Control in Power Systems*. Boston, MA: Springer Science + Business Media, 2005.

[142] D. Arthur and S. Vassilvitskii. “k-means++: The advantages of careful seeding”. In: *Proc. of the 18th annual ACM-SIAM symposium on Discrete algorithms (SODA 2007)*. New Orleans, LA, USA, 2007, pp. 1027–1035.

[143] U. N. Raghavan, R. Albert, and S. Kumara. “Near linear time algorithm to detect community structures in large-scale networks”. In: *Phys. Rev. E* 76.3 (2007), p. 036106.

[144] J. Batson, D. A. Spielman, N. Srivastava, and S.-H. Teng. “Spectral sparsification of graphs: theory and algorithms”. In: *Commun. ACM* 56.8 (2013), pp. 87–94.

[145] X. Xu, N. Yuruk, Z. Feng, and T. A. Schweiger. “Scan: a structural clustering algorithm for networks”. In: *Proc. 13th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2007)*. San Jose, CA, USA, 2007, pp. 824–833.

[146] H. Sun, J. Huang, J. Han, H. Deng, P. Zhao, and B. Feng. “gskeletonclu: Density-based network clustering via structure-connected tree division or agglomeration”. In: *Proc. 10th IEEE International Conference on Data Mining (ICDM 2010)*. Sydney, Australia, 2010, pp. 481–490.

[147] V. Satuluri and S. Parthasarathy. “Symmetrizations for clustering directed graphs”. In: *Proc. 14th International Conference on Extending Database Technology (EDBT 2011)*. ACM. Uppsala, Sweden, 2011, pp. 343–354.

[148] T. Jebara and V. Shchogolev. “B-matching for spectral clustering”. In: *Proc. European Conference on Machine Learning (ECML 2006)*. Ed. by J. F urnkranz, T. Scheffer, and M. Spiliopoulou. Vol. 4212. LNCS. Berlin, Germany: Springer, 2006, pp. 679–686.

[149] B. Yang, V. Vittal, G. T. Heydt, and A. Sen. “A Novel Slow Coherency Based Graph Theoretic Islanding Strategy”. In: *Proc. Power & Energy Society General Meeting (PES GM 2007)*. Tampa, FL, USA, 2007, pp. 1–7.

[150] S. S. Rangapuram and M. Hein. “Constrained 1-Spectral Clustering”. In: *Proc. International conference on Artificial Intelligence and Statistics (AISTATS 2012)*. La Palma, Canary Islands, Spain, 2012, pp. 1–13.

[151] S. Chawla and A. Gionis. “k-means–: A unified approach to clustering and outlier detection”. In: *Proc. 13th SIAM International Conference on Data Mining (SDM 2013)*. Austin, TX, USA, 2013, pp. 189–197.

[152] M. Meila. “Spectral Clustering”. In: *Handbook of cluster analysis*. Ed. by C. Hennig, M. Meila, F. Murtagh, and R. Rocci. CRC Press, 2015, pp. 125–144.

[153] A. Kummerow, S. Nicolai, and P. Bretschneider. “Outlier Detection Methods for Uncovering of Critical Events in Historical Phasor Measurement Records”. In: *3rd International Conference on Power and Renewable Energy (ICPRE 2018)*. Berlin, Germany, 2018, pp. 1–6.

[154] C. C. Aggarwal. *Outlier Analysis*. Springer, 2013.

[155] L. Xiong, X. Chen, and J. Schneider. “Direct robust matrix factorization for anomaly detection”. In: *Proc. 11th IEEE International Conference on Data Mining (ICDM 2011)*. IEEE. 2011, pp. 844–853.

[156] H.-P. Kriegel, A. Zimek, et al. “Angle-based outlier detection in high-dimensional data”. In: *Proc. 14th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2008)*. 2008, pp. 444–452.

[157] U. Brandes, M. Gaertler, and D. Wagner. “Experiments on graph clustering algorithms”. In: *Proc. European Symposium on Algorithms (ESA 2003)*. Ed. by G. Di Battista and U. Zwick. Vol. 2832. LNCS. Springer, Berlin, Heidelberg, 2003, pp. 568–579.

[158] M. Meila and J. Shi. “Learning segmentation by random walks”. In: *Proc. Advances in Neural Information Processing Systems 13 (NIPS 2000)*. Ed. by T. K. Leen, T. G. Dietterich, and V. Tresp. 2001, pp. 873–879.

[159] R. Sedgewick and K. Wayne. *Algorithms, 4th Edition*. Addison-Wesley, 2011.

[160] M. Laszlo and S. Mukherjee. “Minimum spanning tree partitioning algorithm for microaggregation”. In: *IEEE Trans Knowl Data Eng* 17.7 (2005), pp. 902–911.

[161] O. Grygorash, Y. Zhou, and Z. Jorgensen. “Minimum spanning tree based clustering algorithms”. In: *Proc. 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2006)*. 2006, pp. 73–81.

[162] C. Josz, S. Fliscounakis, J. Maeght, and P. Panciatici. “AC power flow data in MATPOWER and QCQP format: iTesla, RTE snapshots, and PEGASE”. In: *arXiv:1603.01533* (2016).

[163] C. Huo and E. Cotilla-Sanchez. “A Power-Balanced Clustering Algorithm to Improve Electrical Infrastructure Resiliency”. In: *Proc. 20th Power Systems Computation Conference (PSCC 2018)*. Dublin, Ireland, 2018, pp. 1–8.

[164] Z. Lin, F. Wen, Y. Ding, and Y. Xue. “Data-Driven Coherency Identification for Generators Based on Spectral Clustering”. In: *IEEE Trans. Ind. Informat.* 14.3 (2018), pp. 1275–1285.

[165] P. J. Rousseeuw. “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis”. In: *J. Comput. Appl. Math* 20 (1987), pp. 53–65.

[166] P. H. Schönemann. “A generalized solution of the orthogonal Procrustes problem”. In: *Psychometrika* 31.1 (1966), pp. 1–10.

[167] J. Verboomen. “Optimisation of Transmission Systems by use of Phase Shifting Transformers”. PhD Thesis. Delft University of Technology, 2008.

[168] M.-F. Balcan, Y. Liang, and P. Gupta. “Robust hierarchical clustering”. In: *J. Mach. Learn. Res.* 15.1 (2014), pp. 3831–3871.

[169] B. Eliasson. “Damping of Power Oscillations in Large Power Systems”. PhD Dissertation. Lund Institute of Technology, 1990.

[170] B. E. Eliasson and D. J. Hill. “Damping structure and sensitivity in the NORDEL power system”. In: *IEEE Trans. Power Syst.* 7.1 (1992), pp. 97–105.

[171] M. Eremia and M. Shahidehpour. *Handbook of Electrical Power System Dynamics: Modeling, Stability, and Control*. John Wiley & Sons, 2013.

[172] B. Stott and O. Alsac. “Fast Decoupled Load Flow”. In: *IEEE Trans. Power App. Syst.* PAS-93.3 (1974), pp. 859–869.

[173] Q. Guo, H. Sun, B. Zhang, and W. Wu. “Power network partitioning based on clustering analysis in Mvar control space”. In: *Autom. Electr. Power Syst.* 29.10 (2005), pp. 36–40.

[174] W. Yan, W. Cui, W.-J. Lee, J. Yu, and X. Zhao. “Pilot-bus-centered automatic voltage control with high penetration level of wind generation”. In: *IEEE Trans. Ind. Appl.* 52.3 (2016), pp. 1962–1969.

[175] H. Ge, Q. Guo, H. Sun, B. Wang, and B. Zhang. “Multivariate statistical analysis-based power-grid-partitioning method”. In: *IET Generation, Transmission & Distribution* 10.4 (2016), pp. 1023–1031.

[176] H.-Y. Su and C.-W. Liu. “An Adaptive PMU-Based Secondary Voltage Control Scheme”. In: *IEEE Trans. Smart Grid* 4.3 (2013), pp. 1514–1522.

[177] A. Stankovic, M. Ilic, and D. Maratukulam. “Recent results in secondary voltage control of power systems”. In: *IEEE Trans. Power Syst.* 6.1 (1991), pp. 94–101.

[178] H. Ge, Q. Guo, H. Sun, B. Wang, B. Zhang, and W. Wu. “A load fluctuation characteristic index and its application to pilot node selection”. In: *Energies* 7.1 (2014), pp. 115–129.

[179] J. L. Sancha, J. L. Fernandez, A. Cortes, and J. T. Abarca. “Secondary voltage control: analysis, solutions and simulation results for the Spanish transmission system”. In: *IEEE Trans. Power Syst.* 11.2 (1996), pp. 630–638.

[180] M. A. Pai. *Energy Function Analysis for Power System Stability*. Boston: Kluwer Academic Publishers, 1989. 240 pp.

[181] D. J. Inman. *Engineering Vibration*. 4th edition. Pearson, 2013.

[182] J. R. Winkelman, J. H. Chow, B. C. Bowler, B. Avramovic, and P. V. Kokotovic. "An analysis of interarea dynamics of multi-machine systems". In: *IEEE Trans. Power App. Syst.* PAS-100.2 (1981), pp. 754–763.

[183] J. H. Chow, R. Galarza, P. Accari, and W. W. Price. "Inertial and slow coherency aggregation algorithms for power system dynamic model reduction". In: *IEEE Trans. Power Syst.* 10.2 (1995), pp. 680–685.

[184] J. Machowski, J. W. Bialek, and J. Bumby. *Power system dynamics: stability and control*. 2nd ed. John Wiley & Sons, 2011.

[185] F. Ma and V. Vittal. "Right-sized power system dynamic equivalents for power system operation". In: *IEEE Trans. Power Syst.* 26.4 (2011), pp. 1998–2005.

[186] W. W. Price, E. M. Gulachenski, P. Kundur, F. J. Lange, G. C. Loehr, B. A. Roth, et al. "Testing of the modal dynamic equivalents technique". In: *IEEE Trans. Power App. Syst.* PAS-97.4 (1978), pp. 1366–1372.

[187] L. Rouco and I. J. Perez-Arriaga. "Multi-area analysis of small signal stability in large electric power systems by SMA". In: *IEEE Trans. Power Syst.* 8.3 (1993), pp. 1257–1265.

[188] G. N. Ramaswamy, L. Rouco, O. Fillatre, G. C. Verghese, P. Panciatici, B. C. Lesieutre, et al. "Synchronic modal equivalencing (SME) for structure-preserving dynamic equivalents". In: *IEEE Trans. Power Syst.* 11.1 (1996), pp. 19–29.

[189] J. Y. Zien, M. D. F. Schlag, and P. K. Chan. "Multilevel spectral hypergraph partitioning with arbitrary vertex sizes". In: *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* 18.9 (1999), pp. 1389–1399.

[190] P. A. Trodden, W. A. Bukhsh, A. Grothey, and K. I. M. McKinnon. "Optimization-Based Islanding of Power Networks Using Piecewise Linear AC Power Flow". In: *IEEE Trans. Power Syst.* 29.3 (2014), pp. 1212–1220.

[191] X. Wang and I. Davidson. "Flexible constrained spectral clustering". In: *Proc. 16th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 2010)*. Washington, DC, USA, 2010, pp. 1–10.

[192] X. Wang, B. Qian, and I. Davidson. "On constrained spectral clustering and its applications". In: *Data Min. Knowl. Discov.* 28.1 (2014), pp. 1–30.

[193] S. D. Kamvar, D. Klein, and C. D. Manning. "Spectral learning". In: *Proc. 18th International Joint Conference on Artificial Intelligence (IJCAI 2003)*. Acapulco, Mexico, 2003, pp. 561–566.

[194] Z. Li, J. Liu, and X. Tang. "Constrained Clustering via Spectral Regularization". In: *Proc. 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*. Miami, FL, USA, 2009, pp. 1–8.

[195] H. P. Williams. *Logic and Integer Programming*. Vol. 130. Internat. Ser. Oper. Res. Management Sci. Springer US, 2009.

[196] T. Ding, K. Sun, C. Huang, Z. Bie, and F. Li. "Mixed-Integer Linear Programming-Based Splitting Strategies for Power System Islanding Operation Considering Network Connectivity". In: *IEEE Syst. J.* 12.1 (2018), pp. 350–359.

[197] A. Kyriacou, P. Demetriou, C. Panayiotou, and E. Kyriakides. "Controlled islanding solution for large-scale power systems". In: *IEEE Trans. Power Syst.* 33.2 (2017), pp. 1591–1602.

[198] B. E. Eliasson and D. J. Hill. "Damping structure and sensitivity in the NORDEL power system". In: *IEEE Trans. Power Syst.* 7.1 (1992), pp. 97–105.

[199] S. Ruder. "An overview of gradient descent optimization algorithms". In: *arXiv:1609.04747* (2016).

[200] Y. E. Nesterov. "A method for solving the convex programming problem with convergence rate $O(1/k^2)$". In: *Dokl. Akad. Nauk SSSR (translated as Soviet Math. Docl.)* 269.3 (1983), pp. 543–547.

[201] I. Sutskever, J. Martens, G. E. Dahl, and G. E. Hinton. "On the importance of initialization and momentum in deep learning". In: *Proc. of the 30th International Conference on Machine Learning (ICML 2013)*. Ed. by S. Dasgupta and D. McAllester. Vol. 28. JMLR: W&CP. Atlanta, GA, USA, 2013, pp. 1–9.

[202] S. Boyd and L. Vandenberghe. *Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares*. 1st edition. Cambridge, UK: Cambridge University Press, 2018.

LIST OF PUBLICATIONS

JOURNAL ARTICLES

[J1] **I. Tyuryukanov**, M. Popov, M. A. M. M. van der Meijden, and V. Terzija. "Discovering Clusters in Power Networks From Orthogonal Structure of Spectral Embedding". In: *IEEE Trans. Power Syst.* 33.6 (2018), pp. 6441–6451.

[J2] **I. Tyuryukanov**, A. C. Karagiannis, M. Popov, M. A. M. M. van der Meijden, and V. Terzija. "Generator grouping cutset determination based on tree construction and constrained spectral clustering". In: *The Journal of Engineering* 2018.15 (2018), pp. 1309–1314.

[J3] **I. Tyuryukanov**, M. Popov, M. A. M. M. van der Meijden, and V. Terzija. "Slow Coherency Identification and Power System Dynamic Model Reduction by using Orthogonal Structure of Electromechanical Eigenvectors". In: *IEEE Trans. Power Syst.* under review (2020).

BOOK CHAPTERS

[B1] **I. Tyuryukanov**, M. Naglič, M. Popov, M. A. M. M. van der Meijden, and V. Terzija. "Controlled Islanding of Power Networks using PowerFactory and MATLAB". In: *Advanced Smart Grid Functionalities based on PowerFactory*. Ed. by F. Gonzalez-Longatt and J. L. Rueda Torres. Springer International Publishing, 2018. Chap. 11, pp. 279–299.

CONFERENCE PAPERS

[C1] **I. Tyuryukanov**, J. Quirós-Tortós, M. Naglič, M. Popov, M. A. M. M. van der Meijden, and V. Terzija. "Controlled Islanding of Power Networks based on Graph Reduction and Spectral Clustering". In: *Proc. 10th Mediterranean Conference on Power Generation, Transmission, Distribution and Energy Conversion (MedPower 2016)*. Belgrade, Serbia, 2016, pp. 1–6.

[C2] **I. Tyuryukanov**, J. Quirós-Tortós, M. Naglič, M. Popov, M. A. M. M. van der Meijden, and V. Terzija. "A Post-processing Methodology for Robust Spectral Embedded Clustering of Power Networks". In: *Proc. 17th IEEE International Conference on Smart Technologies (EUROCON 2017)*. Ohrid, Republic of Macedonia, 2017, pp. 1–5.

[C3] **I. Tyuryukanov**, M. Popov, M. A. M. M. van der Meijden, and V. Terzija. "Spectral MST-based Graph Outlier Detection with Application to Clustering of Power Networks". In: *Proc. 20th Power Systems Computation Conference (PSCC 2018)*. Dublin, Ireland, 2018, pp. 1–8.

OTHER PUBLICATIONS

[E1] **I. Tyuryukanov**, J. Wang, and A. Monti. “Design of a novel robust current controller for grid-connected inverter against grid impedance variations”. In: *International Journal of Electrical Power & Energy Systems* 110 (2019), pp. 454–466.

[E2] M. Naglič, **I. Tyuryukanov**, M. Popov, M. A. M. M. van der Meijden, and V. Terzija. “WAMPAC-ready platform for online evaluation of corrective control algorithms”. In: *Proc. 10th Mediterranean Conference on Power Generation, Transmission, Distribution and Energy Conversion (MedPower 2016)*. Belgrade, Serbia, 2016, pp. 1–6.

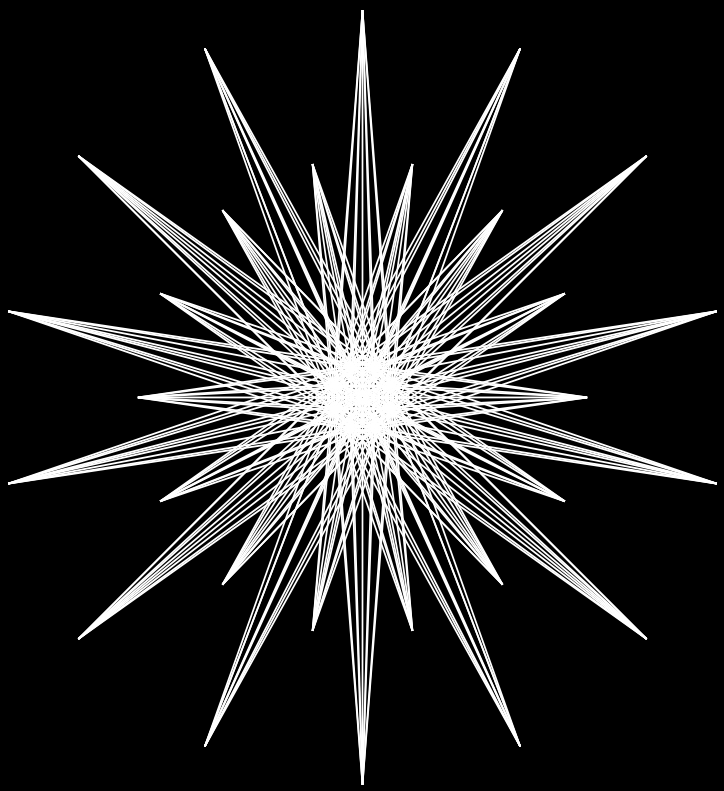
[E3] M. Naglič, L. Liu, **I. Tyuryukanov**, M. Popov, M. A. M. M. van der Meijden, and V. Terzija. “Synchronized Measurement Technology Supported AC and HVDC Online Disturbance Detection”. In: *Proc. International Conference on Power Systems Transients (IPST 2017)*. Seoul, Republic of Korea, 2017, pp. 1–6.

[E4] M. Naglič, L. Liu, **I. Tyuryukanov**, M. Popov, M. A. M. M. van der Meijden, and V. Terzija. “Synchronized measurement technology supported AC and HVDC online disturbance detection”. In: *Electric Power Systems Research* 160 (2018), pp. 308–317.

BIOGRAPHY

Ilya Tyuryukanov was born on August 30 1989 in Moscow Region, Russia. He received the high school diploma from Moscow Lyceum 1502, where he graduated on English, Physics, and Mathematics. In 2006, he enrolled at the Moscow Power Engineering Institute (Technical University) to study protection and control of electric power systems. After graduating with a B.S.E.E. (cum laude) degree in 2010, he participated in a year abroad study program at the TU Ilmenau, Ilmenau, Germany organized by the German Academic Exchange Service (DAAD). In October 2011, he enrolled at RWTH Aachen University, Aachen, Germany on an electrical power engineering M.Sc. program after receiving a one year scholarship from the Carl-Arthur Pastor Foundation. During his master study, he took a six months internship at SMA Solar Technology in the power electronics converter control department and worked part-time for several months as a student assistant. In 2014, he obtained the M.Sc. (cum laude) degree from RWTH Aachen University after defending the thesis titled "Robust AC current control of three-phase inverter-interfaced distributed generator under grid impedance uncertainty". In late 2014, he started as a PhD candidate in the Intelligent Electrical Power Grids (IEPG) group of the Electrical Sustainable Energy (ESE) department of the Delft University of Technology. His research project was related to power system wide-area protection and control (WAMPAC), and it was funded by the Dutch Research Council (NWO) within the program of Uncertainty Reduction of Smart Energy Systems (URSES). His research interests include applying machine learning and optimization techniques to power systems, in particular, wide-area control and protection. Since March 2019 he is working in the same research group as a postdoc researcher within the ESI-bida program (Energie: Systeemintegratie en Big Data).

The vast size of a modern interconnected power grid precludes controlling and operating it as a single object. Subdividing a power grid into a number of internally coherent areas allows to cope with its inherent complexity and to enable more efficient control structures. This thesis focuses on discovering the power system structure to facilitate the definition of control areas for wide-area monitoring, protection and control (WAMPAC) applications. Graph partitioning is a well-developed discipline whose potential is not fully recognized in the power system domain. Particularly, spectral graph partitioning methods are shown to be very promising. Their efficiency is first demonstrated by accurately selecting the number and extent of control zones for secondary voltage control (SVC). Furthermore, it is shown that grouping generators with similar slow rotor angle dynamics is closely related to spectral graph partitioning, and an enhanced slow coherency algorithm is proposed based on these findings. The final topic is constrained graph partitioning subject to node grouping constraints, which is related to intentional controlled islanding (ICI). As both solution time and accuracy are critical for ICI, a new polynomial-time heuristic algorithm is proposed that is more accurate than comparable state-of-the-art methods.



ISBN 978-94-6384-116-0