A Novel Approach to Transmission Power, Lifetime and Connectivity Optimization in Asymmetric Networks

Milad Esmaeilpour

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montréal, Québec, Canada

August 2018

© Milad Esmaeilpour, 2018

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Milad Esmaeilpour

Entitled: A Novel Approach to Transmission Power, Lifetime and Connectivity Optimization in Asymmetric Networks

and submitted in partial fulfilment of the requirements for the degree of

Master of Applied Science

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

______________________ Dr. Hassan Rivaz, Chair

______________________ Dr. Chadi Assi, External Examiner

______________________ Dr. Krzysztof Skonieczny, Examiner

______________________ Dr. Amir Aghdam, Supervisor

Approved by: ______________________

William E. Lynch, Chair
Department of Electrical and Computer Engineering

__________ 2018 ______________________

Amir Asif, Dean
Faculty of Engineering and Computer Science

Abstract

A Novel Approach to Transmission Power, Lifetime and Connectivity Optimization in Asymmetric Networks

Milad Esmaeilpour

This thesis deals with the problem of proper power management over asymmetric networks represented by weighted directed graphs (digraphs) in the presence of various constraints. Three different problems are investigated in this study. First, the problem of total transmission power optimization and connectivity control over the network is examined. The notion of generalized algebraic connectivity (GAC), used as a network connectivity measure, is formulated as an implicit function of the nodes' transmission powers. An optimization problem is then presented to minimize the total transmission power of the network while considering constraints on the values of the GAC and the individual transmission power levels. The problem of network lifetime maximization and connectivity control is investigated afterwards. Each node is assumed to deplete its battery linearly with respect to the transmission powers used for communication, and the network lifetime is defined as the minimum lifetime over all nodes. Finally, it is desired to maximize the connectivity level of the network with constraints on the total transmission power of the network and the individual transmission powers. The interior point and the mixed interior point-exterior point methods are utilized to transform these constrained optimization problems into sequential optimization problems. Given the new formulation, each subproblem is then solved numerically via the subgradient method with backtracking line search. A distributed version of the algorithm, taking into account the estimation of global quantities, is provided. The asymptotic convergence of the proposed centralized and distributed algorithms is demonstrated analytically, and their effectiveness is verified by simulations.

Acknowledgements

I would like to express my gratitude to my supervisor Dr. Amir Aghdam for his immense help and support throughout my master's degree. This thesis would not be possible without his continuous guidance and insightful comments. I am grateful to have had such a knowledgable, considerate and outstanding supervisor.

I would also like to thank Dr. Stephane Blouin for his important comments on the practical aspects of this research.

Thank you to all my colleagues and fellow labmates at Concordia University for their helpful comments and the fun time that we have had together.

Last but not the least, a very special thanks to my parents and my sisters for their unconditional love, encouragement and support throughout my life. None of this would have been possible without them.

List of Publications

[1] M. Esmaeilpour, A. G. Aghdam, and S. Blouin, "Joint transmission power optimization and connectivity control in asymmetric networks," in *Proceedings of the 2018 American Control Conference*, June 2018, to appear.

[2] M. Esmaeilpour, A. G. Aghdam, and S. Blouin, "Lifetime optimization and generalized algebraic connectivity control in asymmetric networks," submitted for conference publication.

[3] M. Esmaeilpour, A. G. Aghdam, and S. Blouin, "Connectivity, transmission power, and lifetime optimization in asymmetric networks: A distributed approach," submitted for journal publication.

Contribution of Authors

This thesis is written in a manuscript-based format, and all the presented papers are entirely written by Milad Esmaeilpour. The papers are co-authored with Dr. Amir Aghdam (Department of Electrical and Computer Engineering, Concordia University) and Dr. Stephane Blouin (Defence Research and Development Canada). As the research supervisor, Dr. Aghdam provided guidance and direction on all aspects of the considered topic. Dr. Blouin gave comments on the practical settings of the networks considered in the thesis, enabling the author to have application-oriented assumptions and simulations. He also reviewed the papers and provided insightful comments. All the research presented in this dissertation is performed and critically analyzed by the author.

Table of Contents

List of Figures . . . . **xi**

List of Tables . . . . **xiv**

Nomenclature . . . . **xv**

1 Introduction **1**

1.1 Motivation 1

1.2 Literature Review 2

1.3 Thesis Contributions 8

1.4 Thesis Layout 10

2 Joint Transmission Power Optimization and Connectivity Control in Asymmetric Networks **12**

2.1 Introduction 14

2.2 Preliminaries 17

2.2.1 Problem Formulation 18

2.3 Convergence Analysis of the Optimization Algorithm 25

2.4 Simulation Results 31

2.5 Conclusion 34

3 Lifetime Optimization and Generalized Algebraic Connectivity Control in Asymmetric Networks 36

3.1 Introduction 38

3.2 Prelimineries 41

3.2.1 Problem Formulation 43

3.3 Convergence Analysis of the Optimization Algorithm 52

3.4 Simulation Results 62

3.5 Conclusion 68

4 Connectivity, Transmission Power, and Lifetime Optimization in Asymmetric Networks: A Distributed Approach 69

4.1 Introduction 71

4.2 Problem Formulation 77

4.2.1 Preliminaries and Notation 77

4.2.2 Optimization Problems 80

4.3 Distributed optimization algorithm 93

4.4 Convergence analysis of the optimization algorithm 109

4.5 Simulation Results 117

4.5.1 Minimizing $\tilde{f}_1$ in (4.7) 119

4.5.2 Minimizing $\tilde{f}_2$ in (4.7) 125

4.5.3 Minimizing $\tilde{f}_3$ in (4.7) 133

4.6 Conclusion 143

5 Conclusions and Future Work 145

5.1 Future Work 148

Bibliography 150

List of Figures

2.1 Evolution of the generalized algebraic connectivity of the network of Example 2.1. . . . 34

2.2 Evolution of the transmission power of every node in the network of Example 2.1. . . . 35

3.1 Evolution of the lifetimes of the nodes of the network of Example 3.1. . . . 66

3.2 Evolution of the transmission power for nodes making up the network of Example 3.1. . . . 67

4.1 The inner optimization loop (line 4 of Algorithm 4.1). . . . 94

4.2 Evolution of the GAC and the total transmission power of the network of Example 4.1 for the first optimization problem in (4.7) ($l = 1$) obtained using the centralized algorithm. . . . 121

4.3 Evolution of the transmission power for the nodes comprising the network of Example 4.1 for the first optimization problem in (4.7) ($l = 1$) obtained using the centralized algorithm. . . . 122

4.4 Evolution of the total transmission power and the GAC of the network of the network of Example 4.1 for the second optimization problem in (4.7) ($l = 2$) obtained using the centralized algorithm. 127

4.5 Evolution of the transmission power for the nodes comprising the network of Example 4.1 for the second optimization problem in (4.7) ($l = 2$) obtained using the centralized optimization algorithm. 128

4.6 Evolution of the total transmission power and the GAC of the network of the network of Example 4.1 for the second optimization problem in (4.7) ($l = 2$) obtained using the distributed algorithm. 130

4.7 Evolution of the transmission power for the nodes comprising the network of Example 4.1 for the second optimization problem in (4.7) ($l = 2$) obtained using the distributed algorithm. 131

4.8 Evolution of the lifetime and the GAC of the network of the network of Example 4.1 for the third optimization problem in (4.7) ($l = 3$) obtained using the centralized algorithm. 135

4.9 Evolution of the lifetime of the nodes of the network of the network of Example 4.1 for the third optimization problem in (4.7) ($l = 3$) obtained using the centralized algorithm. 136

4.10 Evolution of the transmission power for the nodes comprising the network of Example 4.1 for the third optimization problem in (4.7) ($l = 3$) obtained using the centralized algorithm. 137

4.11 Evolution of the lifetime and the GAC of the network of the network of Example 4.1 for the third optimization problem in (4.7) ($l = 3$) obtained using the distributed algorithm. 139

4.12 Evolution of the lifetime of the nodes of the network of the network of Example 4.1 for the third optimization problem in (4.7) ($l = 3$) obtained using the distributed algorithm. 140

4.13 Evolution of the transmission power for the nodes comprising the network of Example 4.1 for the third optimization problem in (4.7) ($l = 3$) obtained using the distributed algorithm. 141

List of Tables

4.1 The proposed optimization problems. 86

Nomenclature

General

$\mathbb{R}_{>r}$	Set of real numbers greater than $r \in \mathbb{R}$
$\mathbb{N}_n$	Finite set of natural numbers $\{1, 2, ..., n\}$
$^{\mathrm{T}}$	Transpose of a real vector or matrix
$\mathrm{tr}(\cdot)$	Trace of a given real matrix
$\langle\cdot,\cdot\rangle$	Inner product of two real vectors or matrices
$\lceil\cdot\rceil$	Ceiling function
$\Re(\cdot)$	Real part of a complex number
$\mathbb{B}_\sigma(\cdot)$	Closed ball of radius $\sigma \in \mathbb{R}_{>0}$ centered at a given point
1_n	All-one vector of length n

e_i Column vector whose elements are all zero, except for its i^{th} element which is equal to one

$\mathbf{e}_{ij}$ Matrix whose elements are all zero, except for its $(i,j)^{\text{th}}$ element which is equal to one

$\|\cdot\|$ Euclidean norm of a given real vector

$\|\cdot\|_{\mathrm{F}}$ Frobenius norm of a given real matrix

k Iteration index of the optimization loop

G Weighted directed graph representing the asymmetric network

n Number of nodes in the network

V Set of vertices of the weighted directed graph G

E Edge set of the weighted directed graph G

$\mathbf{W}$ Weight matrix of the weighted directed graph G

w_{ij} Weight associated with the edge (j,i)

N_i^{in} In-neighbor set associated with node i

N_i^{out} Out-neighbor set associated with node i

$\mathbf{L}$ Laplacian matrix of the weighted directed graph G

$\lambda_i(\cdot)$	i^{th} eigenvalue of a given matrix
$\Lambda(\cdot)$	Spectrum of a given matrix
$\tilde{\lambda}(\cdot)$	Generalized algebraic connectivity of a given weighted directed graph G
$h(\cdot;\cdot)$	The function relating transmission power to communication link weight
ξ_{ij}	Set of real constant parameters characterizing the communication channel (j, i)
$\underline{\lambda}$	Smallest acceptable connectivity level
m	Number of the constraints of the optimization problem
Γ	Coefficient of the penalty function $I(\cdot)$
Γ^0	Initial value of the parameter Γ
ϵ	Accuracy of the optimization algorithm
μ	Factor which Γ is multiplied with
μ_0	Initial value of the parameter μ
α	Step-size to move along the search direction
α_{max}	Initial and maximum value taken by the step-size
β	Step-size used to numerically calculate the partial derivatives of the GAC

ν Expected amount of decrease in the joint cost function moving along the search direction

θ Factor by which the step-size is reduced α

m_{iter} Current iteration of the inner optimization loop

m_{max} Maximum number of the iterations of the inner optimization loop

k' Next iteration of the optimization loop when the search direction at iteration k is not descent

Chapter 2

P Transmission power vector of the network

P_i Transmission power of node i

P_i^{low} Lower bound of the permissible transmission power for the i^{th} node

P_i^{up} Upper bound of the permissible transmission power for the i^{th} node

$\mathcal{P}$ Permissible transmission power set

$\tilde{f}(\cdot,\cdot)$ Joint cost function to be minimized

$\partial\tilde{f}(\cdot,\cdot)$ Subdifferential set of the joint cost function

$I(\cdot)$ Penalty function obtained via the interior point method for the inequality constraints

$\mathcal{R}$ Domain of the joint cost function $\tilde{f}(\cdot,\cdot)$

P^0 Initial transmission power vector

g Arbitrary subgradient vector of the joint cost function

g_i i^{th} element of the arbitrary subgradient vector of the joint cost function

$g_i^{\pm}$ i^{th} element of the subgradient vector of the joint cost function where $\nabla\tilde{\lambda}_i(\cdot) = \frac{1}{\beta}\partial\tilde{\lambda}_i^{\pm}(\cdot)$ respectively

v Search direction for the transmission power vector

v_i i^{th} element of the search direction

$\nabla\tilde{\lambda}_i$ i^{th} element of the supergradient vector of the GAC

$\partial\tilde{\lambda}_i$ i^{th} element of the superdifferential set of the GAC

v^* Global minimum of the considered optimization problem

P^* Minimizer of the optimization problem corresponding to v^*

P^*_Γ Central point of the interior point algorithm for a specific Γ

P_{Γ}^{0} The power vector that the inner optimization loop starts working with when $m_{iter} = 1$ for a specific Γ

Chapter 3

$\tilde{f}(\cdot,\cdot,\cdot)$ Joint cost function to be minimized

$\partial\tilde{f}(\cdot,\cdot,\cdot)$ Subdifferential set of the joint cost function

$\mathcal{R}$ Domain of the joint cost function $\tilde{f}(\cdot,\cdot,\cdot)$

$I(\cdot)$ Penalty function obtained via the interior point method for the inequality constraints

$O(\cdot)$ Penalty function obtained via the exterior point method for the equality constraints

Υ Coefficient of the penalty function $O(\cdot)$

Υ^0 Initial value of the parameter Υ

δ Factor which Υ is multiplied with

δ_0 Initial value of the parameter δ

$\mathrm{P}_{\Gamma,\Upsilon}^{*}$ Central point of the interior point-exterior point algorithm for a specific Γ and Υ

v^* Local minimum of the considered optimization problem

$\mathbf{P}^*$ Minimizer of the optimization problem corresponding to v^*

Chapter 4

$\tilde{f}_l(\cdot,\cdot)$ Joint cost function to be minimized

$\partial\tilde{f}_l(\cdot,\cdot)$ Subdifferential set of the joint cost function

$\mathcal{R}_l$ Domain of the joint cost function $\tilde{f}_l(\cdot,\cdot)$

$I_l(\cdot)$ Penalty function

$\bar{\mathrm{P}}$ Highest acceptable total transmission power

v_l^* Local minimum of the considered optimization problem

$\mathbf{P}_l^*$ Minimizer of the optimization problem corresponding to v_l^*

L Lipschitz constant of the GAC

P_Γ^* Central point of the interior point algorithm for a specific Γ

Chapters 3 & 4

$\mathbf{P}$ Transmission power matrix of the network

$\mathbf{P}^0$ Initial transmission power matrix of the network

P_{ij} Transmission power that node j uses to transmit information to node i

P_{ij}^{low} Lower bound of the permissible transmission power P_{ij}

P_{ij}^{up} Upper bound of the permissible node transmission power P_{ij}

P_i Transmission power vector of node i

$T(\cdot)$ Lifetime of node i

$T_{sys}(\cdot)$ Network Lifetime

$\mathfrak{e}_i^0$ Initial energy available to node i

q_{ji} Transmission rate per unit of time from node i to node j

τ_{ji} Time node i has to keep transmitting one packet of information to ensure it has been received at node j

K_{ji} Number of information packets sent from node i to node j

$\mathfrak{e}_{ij}^r$ Energy used by node i to receive information from node j

$\mathbf{g}$ Arbitrary subgradient matrix of the joint cost function

g_{ij} $(i,j)^{\mathrm{th}}$ element of the arbitrary subgradient matrix of the joint cost function

$g_{ij}^{\pm}$ i^{th} element of the subgradient vector of the joint cost function where $\nabla\tilde{\lambda}_i(\cdot) = \frac{1}{\beta}\partial\tilde{\lambda}_i^{\pm}(\cdot)$ respectively

v_{ij} Search direction for the element P_{ij} of node i

ζ	Threshold below which the step-size can be considered approximately zero

Chapter 1

Introduction

1.1 Motivation

Sensor networks consist of spatially distributed fixed or mobile sensors capable of sensing, processing and exchanging data without the need for a pre-existing framework. The challenges involved in deploying good-performing networks and the recent advances in computation, communication, and sensing have stimulated substantial research in this domain [1, 4, 26, 43, 44]. These networks have a multitude of applications in various fields, e.g., environmental, health care or machine health monitoring, target detection and localization, surveillance, disaster control, smart farming, etc. [15, 18, 20, 25, 26, 43].

In deploying sensor networks, several issues need to be addressed, and in particular, the connectivity of the network and its lifetime are two of the most important issues.

Sensor networks typically utilize distributed and cooperative algorithms in order to determine specific, often-global quantities using only local information [28]. The higher the connectivity level of a network is, the more efficiently it will diffuse the information, resulting in faster convergence of the distributed algorithms [2, 4]. The prerequisite to having higher connectivity is stronger communication links between the sensors, which in a noise-limited environment, results from higher transmission powers used by the nodes for communication with their neighbors [28]. Even though having a highly connected network is desirable, it will be at the expense of higher total transmission power in the network. On the other hand, the network lifetime and its power consumption have an inverse relationship [21], meaning that the higher transmission powers required to have higher connectivity in the network will lead to decreased network lifetime. Incapacitation of some nodes due to premature battery depletion can result in a disconnected network, which in turn will prevent the network from completing its mission. Given the importance of the connectivity and the lifetime of the network, the objective of this work is to determine an appropriate balance between the two.

1.2 Literature Review

Recent years have witnessed immense interest in sensor networks. The communication links of these networks can be represented by a graph, which may either be undirected (symmetric) or directed (asymmetric). In undirected graphs, the communication links between the nodes are bi-directional, whereas they may be uni-directional in directed graphs. An example of a symmetric network is the typical terrestrial wireless sensor networks (WSN), and an example of an asymmetric network, where the communication link between two distinct nodes are often uni-directional, is underwater acoustic sensor networks (UWASN) [15, 28, 55]. In the latter example, some sources of noise and uncertainty include multipath propagation, temperature fluctuations, sound speed profile variations, and nearby shipping activity [15, 36, 55]. Another difference between the terrestrial and underwater sensor networks is that contrary to the WSNs which may consist of hundreds of nodes for a speific application, the number of deployed nodes in UWASNs is much smaller. For instance, the experimental network of [15] consists of only four nodes.

An important aspect of deploying sensor networks is their connectivity as discussed in the previous section. Different connectivity measures are proposed in the literature to capture different operational characteristics of a network. For instance, the vertex (or edge) connectivity of a network is the minimum number of nodes (or communication links) whose deletion disconnects the network [45]. These two measures show the network robustness to node and link failure, respectively, and have been investigated in detail in the literature, e.g., see [46]- [49]. The edge connectivity has been extended to asymmetric networks represented by weighted directed graphs (digraphs) in [1], taking into account the joint effects of path reliability and the network robustness to link failure. The topic of interest in this study is the algebraic connectivity of a network. As mentioned in the previous section, sensor networks need to use distributed algorithms to determine certain global values. It is well-known that the convergence rate of these algorithms is directly related to the algebraic connectivity of the network. In general, a highly connected network diffuses information more efficiently [2, 4, 28]. Additionally, having an algebraic connectivity measure allows one to apply mathematical tools such as differential operators on the considered measure. Algebraic connectivity is introduced in [27] as the second smallest eigenvalue of the Laplacian matrix of the undirected graph representing the network, and has been used as a measure of connectivity in symmetric networks. There are numerous studies investigating algebraic connectivity in symmetric networks. For example, a distributed algorithm is presented in [3] to estimate and control the algebraic connectivity of undirected graphs using a stochastic power iteration method. In [37], a distributed method, relying on the distributed computation of the powers of the adjacency matrix, is proposed to obtain upper and lower bounds at each iteration for the algebraic connectivity of a symmetric network. As the algorithm proceeds, these bounds converge to the true value of the algebraic connectivity. In addition, a supergradient algorithm is used along with a decentralized eigenvector estimation strategy in [5] to maximize the algebraic connectivity of a symmetric network. In [50], centralized and distributed algorithms are proposed to maintain, increase and control connectivity in mobile robot networks, where mobility is used to control the topology of the underlying communication network. The authors of [51] consider a particular event-triggered consensus scenario, and show that show that the availability of an estimate of the algebraic connectivity could be used for adapting the behavior of the average consensus algorithm. A novel distributed algorithm is also presented for estimating the algebraic connectivity which requires the distributed computation of the powers of matrices.

The counterpart of algebraic connectivity in asymmetric networks has not been investigated as much. A simple extension of algebraic connectivity to directed graphs is proposed in [38], where the magnitude of the smallest nonzero eigenvalue of the Laplacian matrix is presented as a measure of connectivity. However, this notion fails to capture any operational characteristic of the network. To address this shortcoming, the notion of the generalized algebraic connectivity (GAC) introduced in [39] as the real part of the smallest nonzero eigenvalue of the Laplacian matrix of the weighted digraph representing the network, and is shown to be directly related to the asymptotic convergence of consensus algorithms running over the network. Note that since the Laplacian matrix of an asymmetric network is also asymmetric, it can have complex eigenvalues. A distributed algorithm based on the subspace consensus approach is proposed in [4] for computation of the GAC values using only local information. Furthermore, in [2], the

GAC is formulated as an implicit function of the transmission powers nodes use for communicating with their neighbors, and then is maximized via a distributed supergradient algorithm.

Another fundamental aspect of a sensor network is the power consumption of its nodes, which directly affects the network lifetime. Power consumption in sensor networks is either communication-related or non-communication-related, where the former contributes the most to power consumption [17]. Sensor nodes are typically battery-powered, and recharging or replacing their batteries is not always a viable option. Incapacitation of some nodes due to battery depletion can result in a disconnected network, which in turn can prevent the network from completing its mission [20, 23, 28]. As a result, an appropriate power management scheme is crucial for the efficient operation of any sensor network. The network lifetime is typically defined as the time it takes for the first node to completely deplete its energy [17, 19, 21, 22]. Numerous studies in the literature consider the network lifetime as an explicit performance index. The authors of [21] consider a routing problem in static wireless ad hoc networks, where the objective is to maximize the network lifetime. They propose a shortest path routing algorithm using link weights that reflect both the communication energy consumption rates and the residual energy levels at the two end nodes of the path. In [19], an optimal control approach is used to solve the problem of routing in sensor networks with the goal of maximizing the network's lifetime. The authors consider a dynamic energy consumption model for the batteries, capturing their nonlinear behavior. In a fixed topology, they show that there exists an optimal policy consisting of time-invariant routing probabilities. The authors extend these results further in [17] where they consider a more general state space battery model. They also consider a joint routing and initial energy allocation problem over the network and prove that the optimal policy depletes the energy reserves of all nodes simultaneously. In [23], base station mobility is proposed as a remedy for countering inefficient routing and topology in WSNs. The authors build a framework to characterize the impact of various mobility patterns on the network lifetime and conclude that optimal Gaussian and spiral patterns result in the highest lifetime values. A mobile sensor network for monitoring a moving target is investigated in [22], where an algorithm is developed to find a near-optimal relocation strategy for the sensors as well as an energy-efficient route for transferring information from the target to destination. The author of [40] propose an optimal distance-based transmission strategy based on ant colony optimization to maximize the lifetime of WSNs and show the effectiveness of their findings by simulations. In [41], the joint optimal design of the physical, medium access control, and network layers is considered to maximize the lifetime of WSNs with limited available energy. The optimization problem is formulated by taking into account several network variables such as the routing flow, transmission rate, etc. The Gauss-Seidel algorithm, in conjunction with the gradient method, is used to update the considered network variables. The authors of [24] provide a mathematical model for network lifetime maximization integrating WSN design decisions on sensor places, activity schedules, data routes and trajectory of the mobile sink(s). They then present two heuristic approaches for the solution of the model and show its efficacy via numerical experiments. For further studies on network lifetime maximization, the interested reader is referred to recent survey studies such as [25] and [26].

1.3 Thesis Contributions

Given the motivation behind this study as discussed in Section 1.1, three optimization problems are investigated in this thesis. In the first problem, it is desired to minimize the total transmission power of the network while ensuring that connectivity level is maintained above a certain level, and that the transmission power values are bounded within prescribed limits. The objective of the second optimization problem is to maximize the lifetime of the network subject to constraints on the values of the GAC and the transmission powers used for communication. The last problem investigates the maximization of the network connectivity, i.e., the GAC, while satisfying constraints on the total transmission power of the network and the individual transmission power values.

First contribution of the current work is the formulation of the considered optimization problems. More specifically, the use of the GAC as the measure of network connectivity in conjunction with the total transmission power of the network and its lifetime can be mentioned. A problem similar to the first optimization problem is considered in [1], where the total transmission power of the network is minimized and is subject to a constraint on the weighted edge connectivity. The considered metric is non-algebraic, whereas in this study, the GAC is considered as the measure of connectivity. In [2], a problem similar to the third optimization problem is considered, where it is desired to maximize the GAC of the network. Unlike this work, [2] does not consider any constraints on the total transmission power of the network. Additionally, the approach of this study and those of [1, 2] in solving the considered optimization problems are different. Notably, in [2], a projection map is used to keep the transmission powers bounded to a pre-defined range, whereas in this study, the constraints are incorporated directly into the cost function to be optimized.

To solve the considered optimization problems, the interior point and the mixed interior point-exterior point methods of [13] are utilized to transform the constrained optimization problems into sequential unconstrained problems. Afterwards, the subgradient method and the backtracking line search are utilized to solve the subproblems. Unlike the gradient method, the subgradient approach does not necessarily generate descent directions at each iteration of the optimization algorithm [42]. Addressing this issue is another contribution of this study. Due to the property of the subgradient method not always being a descent direction, the step-sizes to move along the search directions are typically fixed ahead of time, e.g., see the approach of [2] for determining the step-sizes. In this study on the other hand, since it is ensured that one has a descent search direction at each iteration, the backtracking line search is utilized which allows the calculation of the step-sizes online. Furthermore, in proposing the distributed optimization algorithm, due to the existence of global values such as the GAC which need to be estimated using only local information, an approximate backtracking line search is proposed which does not require the estimation of new GAC values, hence making the proposed algorithm more computationally friendly. This can also be considered as another contribution of the current work.

1.4 Thesis Layout

The structure of the thesis is as follows:

- **Chapter 1** includes the motivation behind this study, the literature review on the connectivity and lifetime of sensor networks, and finally, the contributions of the current work.

- **Chapter 2** investigates the problem of total transmission power optimization and algebraic connectivity control in asymmetric networks using a centralized approach. An optimization algorithm is proposed to numerically solve the considered problem, and its asymptotic convergence is analytically demonstrated. The simulation results show the effectiveness of the proposed algorithm.

- **Chapter 3** investigates the problem of network lifetime optimization and algebraic connectivity control in asymmetric networks using a centralized approach. To numerically solve the considered problem, an optimization algorithm is proposed. The asymptotic convergence of the algorithm is demonstrated analytically, and its effectiveness is verified by simulations.

- **Chapter 4**, in addition to both of the problems of Chapters 2 and 3, investigates the problem of algebraic connectivity optimization and transmission power control in asymmetric networks using a distributed approach. A distributed optimization algorithm is proposed to numerically solve all three optimization problems, taking into account the distributed estimation of global variables. The asymptotic convergence of the proposed algorithm is shown analytically, and its effectiveness is evaluated via numerical simulations.

- **Chapter 5** presents the conclusion as well as the possible directions for future work.

Chapter 2

Joint Transmission Power Optimization and Connectivity Control in Asymmetric Networks

This chapter investigates the problem of transmission power optimization and algebraic connectivity control over asymmetric networks represented by weighted directed graphs (digraphs) using a centralized approach. The notion of generalized algebraic connectivity (GAC), introduced in the literature as a measure of connectivity in weighted digraphs, is formulated as an implicit function of the network's transmission power vector. An optimization problem is then presented to minimize the total transmission power of the network while satisfying constraints on the values of the GAC and the transmission powers. The interior point method is utilized to transform the constrained optimization problem into a sequential unconstrained optimization problem. Each subproblem is solved via the subgradient method with backtracking line search used for step-size calculation. Even though the GAC is a nonconvex and non-differentiable continuous function of the network's transmission power vector, using the aforementioned approach, the optimization problem gradually becomes convex as the algorithm proceeds. Asymptotic convergence of the proposed algorithm is demonstrated analytically, and its effectiveness is verified by simulations.

This chapter is based on the following publication:

M. Esmaeilpour, A. G. Aghdam, and S. Blouin, "Joint transmission power optimization and connectivity control in asymmetric networks," in *Proceedings of the 2018 American Control Conference*, June 2018, to appear.

The above-mentioned manuscript is presented with minimal cosmetic changes in the sequel. The proof for Theorem 2.1, omitted in the conference paper due to space limitations (but presented in detail in the journal paper), is similar to the proof of Theorem 4.1. Additionally, unlike Chapters 3 and 4 where it is assumed that each node uses a different transmission power level to communicate with a subset of its neighbors, in this chapter, it is assumed that the same transmission power level is used for communication with neighbors; hence, the concatenation of all transmission powers is a vector in this chapter, in contrast to a matrix in Chapters 3 and 4.

2.1 Introduction

Sensor networks consist of spatially distributed fixed or mobile sensor nodes, and are used for target localization, parameter estimation, etc. To determine a specific (often-global) quantity from local measurements, sensors need to utilize cooperative algorithms over the network [1]. The convergence rate of such algorithms is directly related to the connectivity level of the network. A network with higher connectivity diffuses the information more efficiently, in general [2]. For symmetric networks represented by undirected graphs, algebraic connectivity is defined as the smallest non-zero eigenvalue of the graph's Laplacian matrix [3]. The counterpart of this measure for asymmetric networks, represented by weighted directed graphs (digraphs), is referred to as the generalized algebraic connectivity (GAC), and is defined as the real part of the smallest non-zero eigenvalue of the Laplacian matrix of the digraph [4]. This measure is shown to be closely related to the convergence rate of the distributed algorithms running on an asymmetric network [4]. Underwater acoustic sensor networks are an example of an asymmetric network with applications in environmental monitoring, underwater exploration, etc. [1].

The connectivity control problem in symmetric networks has been investigated thoroughly in the literature. For instance, in [5], a supergradient algorithm is employed in conjunction with a decentralized strategy for eigenvector computation to maximize algebraic connectivity of a symmetric network. For a network of mobile robots, the authors in [6] propose algorithms for algebraic connectivity maximization using subgradient descent methods as well as network topology control via potential fields. Furthermore, in the case of a random topology, a distributed stochastic power iteration method is first introduced in [3] to estimate the value of algebraic connectivity locally. The resultant estimate is then maximized in the presence of medium access control (MAC) protocols. The authors in [2] investigate the maximization of the GAC for asymmetric networks, where a discrete-time supergradient algorithm is proposed to compute a local maximum of the GAC of the weighted digraph representing the network.

Even though it is normally desirable to have a highly connected sensor network, it would be at the expense of higher total transmission power. In general, this would impose a limit on the lifetime of networks (which are typically battery-powered) [7]. As a result, an appropriate balance between the total transmission power of the network and its connectivity level is of utmost importance. To this end, the optimization problem considered in this chapter aims to minimize the total transmission power of an asymmetric network, while satisfying certain constraints on the values of the GAC and transmission power vector. For symmetric networks, there are several papers in the literature addressing similar optimization problems, e.g. see [8]- [12]. In [8], the authors find the critical power that each node needs to transmit in order to maintain network connectivity. In the presence of node mobility in wireless ad hoc networks, it is shown in [9] that there exists an optimum transmission range maximizing network capacity. The authors of [10] consider the problem of adjusting the transmission powers of a multi-hop wireless sensor network to create a desired topology. They propose two centralized algorithms for use in static networks and two distributed heuristics for mobile networks. In the context of asymmetric networks, only [1] considers the problem of minimizing transmission power subject to a constraint on connectivity. Nonetheless, [1] considers weighted edge connectivity as the measure of network connectivity, which is not algebraic.

In this chapter, the GAC is used as a measure of asymmetric network connectivity in the constrained optimization problem. The interior point method is utilized to convert the resultant nonlinear constrained optimization problem into a sequential unconstrained optimization problem. Due to its simplicity and ability to handle non-differentiability of the GAC, the subgradient method with backtracking line search is employed afterwards to solve each subproblem. Since the subgradient approach may not result in a descent direction in every optimization iteration, a novel technique is also proposed in the present work to address this issue. Asymptotic convergence of the proposed algorithm to the global minimum of the original problem is proved accordingly, and its effectiveness is shown by numerical simulations, where an experimental underwater acoustic sensor network is considered as an example of an asymmetric network.

The remainder of the chapter is organized as follows. In Section 2.2, first, notations and preliminary graph theory concepts used throughout the chapter are given, followed by the optimization problem and its numerical solution. In Section 2.3, convergence analysis of the proposed algorithm is presented. The simulation results are subsequently provided in Section 2.4, and finally, Section 2.5 contains the concluding remarks.

2.2 Preliminaries

Notation: Throughout this chapter, the set of real numbers greater than r is denoted by $\mathbb{R}_{>r}$, and the finite set of natural numbers $\{1, 2, ..., n\}$ is denoted by $\mathbb{N}_n$. The superscript T is used to indicate the transpose of a real vector. Moreover, the inner product of two real vectors $\mathrm{v},\mathrm{w} \in \mathbb{R}^n$ is represented by $\langle \mathrm{v}, \mathrm{w} \rangle$. The real part of a complex number $c \in \mathbb{C}$ is denoted by $\Re(c)$. $\|.\|$ and $\lceil . \rceil$ denote the Euclidean norm and the ceiling function, respectively. Additionally, for a real vector $\mathrm{v} \in \mathbb{R}^n$, $\mathbb{B}_\sigma(\mathrm{v})$ is a closed ball of radius $\sigma \in \mathbb{R}_{>0}$ centered at v, i.e., $\mathbb{B}_\sigma(\mathrm{v}) = \{\mathrm{w} \in \mathbb{R}^n \mid \|\mathrm{w} - \mathrm{v}\| \leq \sigma\}$. Moreover, 1_n is an all-one vector of length n, and $\mathrm{e}_i \in \mathbb{R}^n$ is a column vector whose elements are all zero, except for its i^{th} element which is equal to one.

For any $k \in \mathbb{N}$, let $G(k) = (V, E(k), \mathbf{W}(k))$ denote a weighted directed graph (digraph) in the time interval $[t_k, t_{k+1})$, characterized by a set of vertices $V = \mathbb{N}_n$, a set of edges $E(k)$, and a weight matrix $\mathbf{W}(k) \in \mathbb{R}^{n \times n}$. Note that $\vec{ij} \in E(k)$ if node j receives information from node i in the time interval $[t_k, t_{k+1})$ for any pair of distinct nodes $i, j \in \mathbb{N}_n$ and any $k \in \mathbb{N}$. The (i, j) element of the weight matrix $\mathbf{W}(k)$, denoted by $w_{ij}(k)$, is the weight associated with the link $\vec{ji} \in E(k)$ for any pair of distinct nodes $i, j \in \mathbb{N}_n$ and any $k \in \mathbb{N}$. Furthermore, in the time interval $[t_k, t_{k+1})$, the out-neighbor set associated with node i, the Laplacian of the weighted digraph $G(k)$, and the spectrum of the Laplacian matrix are denoted by $N_i^{out}(k)$, $\mathbf{L}(k) \in \mathbb{R}^{n \times n}$, and $\Lambda(\mathbf{L}(k))$, respectively.

The generalized algebraic connectivity (GAC) of a weighted digraph $G(k)$ with Laplacian matrix $\mathbf{L}(k)$ is defined as the smallest real part of the nonzero eigenvalues of $\mathbf{L}(k)$, i.e.,

$$\tilde{\lambda}(\mathbf{L}(k)) = \min_{\lambda_i(\mathbf{L}(k)) \neq 0,\ \lambda_i(\mathbf{L}(k)) \in \Lambda(\mathbf{L}(k))} \Re(\lambda_i(\mathbf{L}(k))), \tag{2.1}$$

for any $k \in \mathbb{N}$ [4]. It is to be noted that $\tilde{\lambda}(\mathbf{L}(k))$ is a nonconvex and non-differentiable continuous function of the elements of the Laplacian matrix. In contrast to the notion of algebraic connectivity for undirected graphs [3], it is shown in [16] that an increase in the elements of the weight matrix $\mathbf{W}(k)$ of the digraph does not necessarily lead to an increase in the value of the GAC.

2.2.1 Problem Formulation

Consider a time-varying asymmetric network with n stationary nodes, whose information exchange topology is represented by the weighted digraph $G(k)$ for all $k \in \mathbb{N}$, as noted earlier. The transmission power vector of the network is denoted by $\mathrm{P}(k) = [P_1(k), \ldots, P_n(k)]^{\mathrm{T}} \in \mathbb{R}^n$, where $P_i(k) \in [P_i^{\mathrm{low}}, P_i^{\mathrm{up}}]$ is the transmission power of the i^{th} node for any $i \in \mathbb{N}_n$ and $k \in \mathbb{N}$. Furthermore, $\mathrm{P}(k) \in \mathcal{P}$ for any $k \in \mathbb{N}$, where $\mathcal{P} = \prod_{i=1}^{n} [P_i^{\mathrm{low}}, P_i^{\mathrm{up}}] \subset \mathbb{R}^n$ is a compact and convex set [2]. The relation between the i^{th} node's transmission power $P_i(k)$ and the link weight $w_{ij}(k)$ can be described by a function of the following form

$$w_{ij}(k) = h(P_i(k); \xi_{ij}), \tag{2.2}$$

for any $i \in \mathbb{N}_n$, $j \in N_i^{out}(k)$, and $k \in \mathbb{N}$, where ξ_{ij} represents a set of real constant parameters characterizing the communication channel $\vec{ji}$, and $h(.;.)$ is an increasing continuous function [1]. The value of the transmission power of any node directly impacts the weights of its outgoing links. Note that with the above formulation, the GAC of the network can now be expressed as $\tilde{\lambda}(\mathrm{P}(k))$, an implicit function of the transmission power vector, for any $k \in \mathbb{N}$.

In general, a higher weight $w_{ij}(k)$ implies a stronger communication link $\vec{ji}$ at the cost of a higher power consumption by node i in the time interval $[t_k, t_{k+1})$. Given that an appropriate balance between the total transmission power of the network and its connectivity level is imperative, the following optimization problem is considered for the network

$$\begin{aligned} \underset{\mathrm{P}}{\text{minimize}} \quad & \sum_{i=1}^{n} P_i \\ \text{subject to} \quad & \tilde{\lambda}(\mathrm{P}(k)) \geq \underline{\lambda}, \\ & P_i^{\text{low}} \leq P_i(k) \leq P_i^{\text{up}}, \end{aligned} \tag{2.3}$$

for all $i \in \mathbb{N}_n$ and $k \in \mathbb{N}$, where $\underline{\lambda}$ is a prespecified constant, reflecting the smallest acceptable connectivity level, and P_i^{low} and P_i^{up} are, respectively, the fixed lower bound and upper bound of the permissible transmission power for the i^{th} node, which are known *a priori*. Since $\tilde{\lambda}(\mathrm{P}(k))$ is a nonlinear function of the transmission power vector, and also the number of constraints $m = 1 + 2n$ increases linearly with the size of the network, finding an analytical solution to the optimization problem given above may not be feasible. Hence, a particular interior point algorithm, called the logarithmic barrier method, is used in the sequel to numerically tackle the problem.

From (2.3), for any $i \in \mathbb{N}_n$ and $k \in \mathbb{N}$, define

$$f(\mathrm{P}(k)) = \sum_{i=1}^{n} P_i(k)$$

as the main cost function, and

$$h^1(\mathrm{P}(k)) = \underline{\lambda} - \tilde{\lambda}(\mathrm{P}(k)),$$

$$h_i^2(P_i(k)) = P_i(k) - P_i^{\mathrm{up}},$$

$$h_i^3(P_i(k)) = P_i^{\mathrm{low}} - P_i(k),$$

as the constraint functions. For any $i \in \mathbb{N}_n$ and $k \in \mathbb{N}$, functions $f(\mathrm{P}(k))$, $h_i^2(P_i(k))$ and $h_i^3(P_i(k))$ are continuous and differentiable, whereas $h^1(\mathrm{P}(k))$ is a non-differentiable continuous function, and is also nonconvex. The objective now is to transform the constrained optimization problem (2.3) into a sequential unconstrained optimization problem using the logarithmic barrier method. To this end, it is noted that problem

(2.3) has the same minimizer as the following problem

$$\underset{\mathrm{P}}{\text{minimize}} \quad \tilde{f}(\mathrm{P}, \Gamma), \tag{2.4}$$

where

$$\tilde{f}(\mathrm{P}(k), \Gamma) = f(\mathrm{P}(k)) + \Gamma^{-1} I(\mathrm{P}(k)), \tag{2.5}$$

and

$$I(\mathrm{P}(k)) = -\log\left(-h^1(\mathrm{P}(k))\right) - \sum_{i=1}^{n} \log\left(-h_i^2(P_i(k))\right) - \sum_{i=1}^{n} \log\left(-h_i^3(P_i(k))\right) \tag{2.6}$$

for any $k \in \mathbb{N}$ and $\Gamma \in \mathbb{R}_{>0}$ [13, 14]. Define the set $\mathcal{R} \subset \mathcal{P}$ as the domain of function $\tilde{f}(\mathrm{P}(k), \Gamma)$, called hereafter the joint cost function. The procedure to numerically solve the optimization problem (2.4) is given in Algorithm 2.1, where the initial transmission power vector of the algorithm needs to be *strictly feasible*, i.e. P^0 needs to be strictly inside the feasible set $\mathcal{R}$ and not on its boundaries. There are three parameters ϵ, μ and Γ^0 in the algorithm that need to be chosen appropriately. The choice of ϵ involves a trade-off between the accuracy of the method and its execution speed. The parameter μ, on the other hand, determines the rate of increase of Γ at each iteration, and will be discussed later. The parameter Γ^0 determines the initial weight given to the logarithmic penalty function $I(\mathrm{P}(k))$. If the solution to problem (2.4) lies on the boundaries of the set $\mathcal{R}$, we will have $\lim_{k\to\infty} I(\mathrm{P}(k)) = \infty$, whereas $\lim_{k\to\infty} f(\mathrm{P}(k)) \in \mathbb{R}$. To ensure

Algorithm 2.1. Interior point method

1: Given strictly feasible $\mathrm{P} = \mathrm{P}^0 \in \mathbb{R}^n$, initialize $\Gamma = \Gamma^0 \in \mathbb{R}_{>0}$.
2: Choose arbitrary constants $\mu \in \mathbb{R}_{>1}$ and $\epsilon \in \mathbb{R}_{>0}$.
3: **while** $m\Gamma^{-1} > \epsilon$
4: Compute $\mathrm{P}^* \in \underset{\mathrm{P}\in\mathcal{R}}{\operatorname{argmin}}\ \tilde{f}(\mathrm{P}, \Gamma)$.
5: $\mathrm{P} = \mathrm{P}^*$
6: $\Gamma = \mu\Gamma$
7: **end while**

that the solution of the optimization problem (2.4), as $k \to \infty$, is not swayed towards mainly minimizing $I(\mathrm{P}(k))$, rather than $f(\mathrm{P}(k))$, finding proper values for μ and Γ^0 is imperative.

Since (2.4) is a sequential optimization problem, in order to solve each subproblem (line 4 of Algorithm 2.1), the subgradient method is utilized. The definition of a function's subgradient is given next.

Definition 2.1. Vector g is said to be the subgradient of the nonconvex and non-differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ at $\mathrm{x} \in \mathrm{dom}(f)$ if there exists a real scalar $\sigma \in \mathbb{R}_{>0}$ such that

$$f(\mathrm{y}) \geq f(\mathrm{x}) + \langle \mathrm{y} - \mathrm{x}, \mathrm{g}(\mathrm{x}) \rangle, \tag{2.7}$$

for any $\mathrm{y} \in \mathbb{B}_\sigma(\mathrm{x})$, where $\mathrm{dom}(f)$ denotes the domain of function f. The set of all subgradients of function f at $\mathrm{x} \in \mathrm{dom}(f)$ is called the subdifferential set $\partial f(\mathrm{x})$ [2] .

The subgradient method moves the current iteration of the optimization loop in the opposite direction of a subgradient of the function to be optimized ($\tilde{f}(\mathrm{P}(k), \Gamma)$ in this case). This is described by

$$\mathrm{P}(k+1) = \mathrm{P}(k) + \alpha(k) \cdot \mathrm{v}(k), \tag{2.8}$$

where for any $k \in \mathbb{N}$, $\mathrm{v}(k) = -\mathrm{g}(k)$ is the search direction in the time interval $[t_k, t_{k+1})$, and $\mathrm{g}(k) = [g_1(k), g_2(k), ..., g_n(k)] \in \partial \tilde{f}(\mathrm{P}(k), \Gamma)$ is an arbitrary subgradient of the joint cost function $\tilde{f}$. Also, for any $k \in \mathbb{N}$, $\alpha(k)$ is the step-size, determining how much to move the current iteration along the search direction. For any $i \in \mathbb{N}_n$ and $k \in \mathbb{N}$, the i^{th} element of $\mathrm{g}(k)$ is given by

$$g_i(k) = 1 + \Gamma^{-1}\left(-\frac{\nabla \tilde{\lambda}_i(\mathrm{P}(k))}{\tilde{\lambda}(\mathrm{P}(k)) - \underline{\lambda}} + \frac{1}{P_i^{\mathrm{up}} - P_i(k)} - \frac{1}{P_i(k) - P_i^{\mathrm{low}}}\right), \tag{2.9}$$

where $\nabla \tilde{\lambda}_i(\mathrm{P}(k))$ is the i^{th} element of the supergradient vector of the GAC, and can be chosen arbitrarily from the superdifferential set [2]

$$\partial \tilde{\lambda}_i(\mathrm{P}(k)) \coloneqq \frac{1}{\beta}\{\partial \tilde{\lambda}_i^{+}(\mathrm{P}(k)), \partial \tilde{\lambda}_i^{-}(\mathrm{P}(k))\},$$

with

$$\partial \tilde{\lambda}_i^{+}(\mathrm{P}(k)) = \tilde{\lambda}(\mathrm{P}(k) + \beta \mathrm{e}_i) - \tilde{\lambda}(\mathrm{P}(k)), \tag{2.10a}$$

$$\partial \tilde{\lambda}_i^{-}(\mathrm{P}(k)) = \tilde{\lambda}(\mathrm{P}(k)) - \tilde{\lambda}(\mathrm{P}(k) - \beta \mathrm{e}_i), \tag{2.10b}$$

for some constant $\beta \in \mathbb{R}_{>0}$. Let $\mathrm{g}^{+}(k)$ and $\mathrm{g}^{-}(k)$, respectively, correspond to the cases where $\nabla \tilde{\lambda}_i(\mathrm{P}(k)) = \partial \tilde{\lambda}_i^{+}(\mathrm{P}(k))$ and $\nabla \tilde{\lambda}_i(\mathrm{P}(k)) = \partial \tilde{\lambda}_i^{-}(\mathrm{P}(k))$ in (2.9).

The parameter $\alpha(k)$ is chosen using the backtracking line search method, which provides the maximum allowable step-size to move along a given search direction. This search method starts with a relatively large estimate of the step-size, α_{max}, and iteratively reduces the step-size until a sufficient decrease in the joint cost function $\tilde{f}$ is observed. The proposed algorithm for computing the solution of the optimization problem (2.4), which is a combination of Algorithm 2.1 and the subgradient method with backtracking line search, is presented in Algorithm 2.2. In lines 11 to 14 of the algorithm, the step-size is chosen such that the value of the function in the current iteration decreases as much as possible along the search direction while staying within the feasible set $\mathcal{R}$. Unlike the gradient method, the subgradient technique is not necessarily a descent direction for every $k \in \mathbb{N}$. For the case where $\mathrm{v}(k)$ is not a descent direction in the time interval $[t_k, t_{k+1})$ (as specified later in Lemma 2.1), first, μ (the ratio by which Γ is increased) is updated such that it satisfies the inequality given later in Lemma 3.1, and then the inner optimization loop ends (lines 8 to 10 of Algorithm 2.2). This ensures that the search direction in the next iteration of the optimization loop will be a descent direction.

Assumption 2.1. *The network digraph is assumed to be strongly connected at all times, meaning that there is a directed path from every node in the graph to every other node.*

Assumption 2.2. *It is assumed that as the elements of transmission power vector $P(k)$ vary within the permissible set $\mathcal{P}$ for any $k \in \mathbb{N}$, the edge set of the weighted digraph*

$G(k)$ remains static, i.e., no edges are added or removed during the evolution of the network.

Lemma 2.1. *For the non-differentiable function $\tilde{f}(P(k), \Gamma)$ with subdifferential set $\partial \tilde{f}(P(k), \Gamma)$ at $P(k) \in \mathcal{R}$, $v(k)$ is not a descent direction if*

$$\exists \; g(k) \in \partial \tilde{f}(P(k), \Gamma) \text{ such that } \langle v(k), g(k) \rangle \geq 0, \tag{2.11}$$

for any $k \in \mathbb{N}$ and $\Gamma \in \mathbb{R}_{>0}$.

Proof. The proof follows directly from the definition of the subgradient (Definition 2.1).

■

2.3 Convergence Analysis of the Optimization Algorithm

The asymptotic convergence of the proposed optimization algorithm to the global minimum of the constrained optimization problem (2.3) is provided in this section.

Lemma 2.2. *Consider an asymmetric network composed of n nodes represented by a weighted digraph, and let Assumptions 4.1 and 4.2 hold. Functions $f(P(k))$, $h^1(P(k))$, $h_i^2(P_i(k))$, $h_i^3(P_i(k))$, and $I(P(k))$ of the optimization problem (2.4) satisfy the conditions*

Algorithm 2.2. Sequential unconstrained optimization

Given strictly feasible $\mathrm{P} = \mathrm{P}^0 \in \mathbb{R}^n$, initialize
$\Gamma = \Gamma^0$ and $k = 1$.
Choose arbitrary constants $\nu \in (0,1)$, $\theta \in (0,1)$,
$\mu_0 \in \mathbb{R}_{>1}$, ϵ, $\alpha_{max} \in \mathbb{R}_{>0}$, $m_{max} \in \mathbb{N}$ and consider
the prescribed parameters $\underline{\lambda} \in \mathbb{R}_{>0}$, and P^{up}, $\mathrm{P}^{\mathrm{low}} \in \mathbb{R}^n$.
while $m\Gamma^{-1} > \epsilon$ **do**
  $\mu = \mu_0$
  for $m_{iter} = 1 : m_{max}$ **do**
    Compute $\mathrm{v}(k)$.
    Compute Q_k according to (2.13).
    if $\exists \mathrm{g}(k) \in \partial \tilde{f}(\mathrm{P}(k), \Gamma)$ **such that** $\langle \mathrm{v}(k), \mathrm{g}(k) \rangle \geq 0$ **do**
      $\mu = \max\{\mu_0, \lceil \frac{1}{n} Q_k \rceil\}$
      break
    $\alpha(k) = \alpha_{max}$
    while $\mathrm{P}(k) + \alpha(k) \cdot \mathrm{v}(k) \notin \mathcal{R}$ **or**
  $\tilde{f}(\mathrm{P}(k) + \alpha(k) \cdot \mathrm{v}(k), \Gamma) > \tilde{f}(\mathrm{P}(k), \Gamma) + \nu \cdot \alpha(k) \cdot \langle \mathrm{v}(k), \mathrm{g}(k) \rangle$ **do**
      $\alpha(k) = \theta \cdot \alpha(k)$
    end while
    $\mathrm{P}(k+1) = \mathrm{P}(k) + \alpha(k) \cdot \mathrm{v}(k)$
    $k = k + 1$
  end
  $\Gamma = \mu\Gamma$
end while

of [13, Theorem 8], for all $i \in \mathbb{N}_n$ and $k \in \mathbb{N}$. Hence, using the interior point algorithm, the following relations hold

1. $\lim_{k\to\infty} f(P(k)) = v^*$,
2. $\lim_{k\to\infty} \tilde{f}(P(k), \Gamma) = v^*$,
3. $\lim_{k\to\infty} \Gamma^{-1} I(P(k)) = 0$,
4. $\lim_{k\to\infty} P(k) = P^*$,

where v^* is the global minimum of the optimization problem (2.3), and $P^* = [P_1^*, \ldots, P_n^*]$ is its corresponding minimizer.

Proof. The proof is a straightforward extension of the result in [13, Theorem 8], and is omitted due to space limitations. ■

Lemma 2.2 shows that using the interior point algorithm, it is guaranteed that a sequence P(k) exists for problem (2.4) which converges to the global minimizer of optimization problem (2.3). It is desired now to show that the subgradient method with backtracking line search generates this sequence. To this end, the following lemma and theorem are presented.

Assume that v(k) is not a descent direction for some $k \in \mathbb{N}$. According to Algorithm 2.2, the optimization algorithm ends at iteration k without updating the transmission power vector, and the value of Γ is increased. It is to be noted that the iteration index k does not increase, but rather the optimization loop in the time interval $[t_k, t_{k+1})$ is repeated. For the sake of convergence analysis, this new iteration index is denoted by k', where $k' \neq k+1$, and the corresponding optimization parameter is denoted by $\Gamma' = \mu\Gamma$.

Lemma 2.3. *If $v(k) = -g(k)$ is not a descent direction for some $k \in \mathbb{N}$ according to Lemma 2.1, it will be a descent direction at iteration k', if*

$$\mu \geq \lceil \frac{1}{n} Q_k - 1 \rceil, \tag{2.12}$$

where Q_k is given by

$$Q_k = \frac{1}{\Gamma} \sum_{i=1}^{n} \frac{\partial\tilde{\lambda}_i^-(P(k)) + \partial\tilde{\lambda}_i^+(P(k))}{\tilde{\lambda}(P(k)) - \underline{\lambda}} + \frac{2}{\Gamma} \sum_{i=1}^{n} \frac{P_i^{up} + P_i^{low} - 2P_i(k)}{(P_i^{up} - P_i(k))(P_i(k) - P_i^{low})}. \tag{2.13}$$

Proof. Without loss of generality, assume

$$v_i(k) = -g_i^-(k) = -1 - \frac{1}{\Gamma}[-\frac{\partial\tilde{\lambda}_i^-(\mathrm{P}(k))}{\tilde{\lambda}(\mathrm{P}(k)) - \underline{\lambda}} + \frac{1}{P_i^{\mathrm{up}} - P_i(k)} - \frac{1}{P_i(k) - P_i^{\mathrm{low}}}]. \tag{2.14}$$

for all $i \in \mathbb{N}_n$ and any $k \in \mathbb{N}$. If $\exists \mathrm{g}(k) \in \partial\tilde{f}(\mathrm{P}(k), \Gamma)$ such that $\langle \mathrm{v}(k), \mathrm{g}(k) \rangle \geq 0$, meaning $\mathrm{v}(k)$ is not a descent direction, and given that $\langle \mathrm{v}(k), \mathrm{g}^-(k) \rangle < 0$, we have

$$\begin{aligned}\langle \mathrm{v}(k), \mathrm{g}^+(k) \rangle = \sum_{i=1}^{n} \Big((1 + \frac{1}{\Gamma}[-\frac{\partial\tilde{\lambda}_i^+(\mathrm{P}(k))}{\tilde{\lambda}(\mathrm{P}(k)) - \underline{\lambda}} + \frac{1}{P_i^{\mathrm{up}} - P_i(k)} - \frac{1}{P_i(k) - P_i^{\mathrm{low}}}]) \times \\ (-1 - \frac{1}{\Gamma}[-\frac{\partial\tilde{\lambda}_i^-(\mathrm{P}(k))}{\tilde{\lambda}(\mathrm{P}(k)) - \underline{\lambda}} + \frac{1}{P_i^{\mathrm{up}} - P_i(k)} - \frac{1}{P_i(k) - P_i^{\mathrm{low}}}]) \Big) \geq 0.\end{aligned} \tag{2.15}$$

Define

$$M_1 = \frac{\partial \tilde{\lambda}_i^-(\mathrm{P}(k))}{\tilde{\lambda}(\mathrm{P}(k)) - \underline{\lambda}} - \frac{1}{P_i^{\mathrm{up}} - P_i(k)} + \frac{1}{P_i(k) - P_i^{\mathrm{low}}},$$

and

$$M_2 = \frac{\partial \tilde{\lambda}_i^+(\mathrm{P}(k))}{\tilde{\lambda}(\mathrm{P}(k)) - \underline{\lambda}} - \frac{1}{P_i^{\mathrm{up}} - P_i(k)} + \frac{1}{P_i(k) - P_i^{\mathrm{low}}}.$$

Substituting M_1 and M_2 into (2.15) and simplifying the resultant equation, we have

$$\langle \mathrm{v}(k), \mathrm{g}^+(k) \rangle = -n + \frac{1}{\Gamma} \sum_{i=1}^{n} (M_1 + M_2) - \frac{1}{\Gamma^2} \sum_{i=1}^{n} M_1 \cdot M_2 \geq 0. \tag{2.16}$$

Using a similar procedure in the next iteration yields

$$\langle \mathrm{v}(k'), \mathrm{g}^+(k') \rangle = -n + \frac{1}{\mu \cdot \Gamma} \sum_{i=1}^{n} (M_1 + M_2) - \frac{1}{\mu^2 \cdot \Gamma^2} \sum_{i=1}^{n} M_1 \cdot M_2. \tag{2.17}$$

Since β is assumed to be constant, and given that the transmission power vector P is not updated at iteration k, the values of M_1 and M_2 will remain unchanged from iteration k to k'. Rearranging the right-hand side of (2.16) results in

$$\begin{aligned} &-n + \frac{1}{\mu \cdot \Gamma} \sum_{i=1}^{n} (M_1 + M_2) - \frac{1}{\mu^2 \cdot \Gamma^2} \sum_{i=1}^{n} M_1 \cdot M_2 \\ &\qquad \leq \frac{\mu + 1}{\mu^2} \cdot \frac{1}{\Gamma} \sum_{i=1}^{n} (M_1 + M_2) - (\frac{\mu^2 + 1}{\mu^2}) n - \frac{2}{\mu^2 \cdot \Gamma^2} \sum_{i=1}^{n} M_1 \cdot M_2. \end{aligned} \tag{2.18}$$

It is obvious that the left-hand side of (2.18) is equal to (2.17). Now, considering the right-hand side of (2.18), if

$$\mu + 1 \cdot \frac{1}{\Gamma} \sum_{i=1}^{n} (M_1 + M_2) - (\mu^2 + 1)n - \frac{2}{\Gamma^2} \sum_{i=1}^{n} M_1 \cdot M_2 < 0, \tag{2.19}$$

then it is guaranteed that $\langle \mathrm{v}(k'), \mathrm{g}^+(k') \rangle < 0$. Simplifying (2.19), one arrives at

$$\mu \geq \lceil \frac{1}{n} Q_k - 1 \rceil, \tag{2.20}$$

where

$$Q_k = \frac{1}{\Gamma} \sum_{i=1}^{n} (M_1 + M_2) = \frac{1}{\Gamma} \sum_{i=1}^{n} \frac{\partial \tilde{\lambda}_i^-(\mathrm{P}(k)) + \partial \tilde{\lambda}_i^+(\mathrm{P}(k))}{\tilde{\lambda}(\mathrm{P}(k)) - \underline{\lambda}} + \\ \frac{2}{\Gamma} \sum_{i=1}^{n} \frac{P_i^{\mathrm{up}} + P_i^{\mathrm{low}} - 2P_i(k)}{(P_i^{\mathrm{up}} - P_i(k))(P_i(k) - P_i^{\mathrm{low}})}. \tag{2.21}$$

This completes the proof. ■

Remark 2.1. If for any $k \in \mathbb{N}$, $\mathrm{v}(k)$ is not a descent direction, the optimization in the time interval $[t_k, t_{k+1})$ ends (line 10 of Algorithm 2.2), and is repeated in the next time interval with an updated Γ such that $\mathrm{v}(k)$ becomes a descent direction according to Lemma 3.1. Thus, it can be assumed that the final $\mathrm{v}(k)$, for any $k \in \mathbb{N}$, is always a descent direction.

For each $\Gamma \in \mathbb{R}_{>0}$, over the compact and convex set $\mathcal{R}$, problem (2.4) has a unique global minimum. Let the minimizer corresponding to this minimum be denoted by P_Γ^*, called the central points of the interior point algorithm [14]. Also, let P_Γ^0 denote the power vector that the inner optimization loop starts working with (when $m_{iter} = 1$), for any $\Gamma \in \mathbb{R}_{>0}$. Starting from P_{Γ}^{0}, it is desired to make the $\tilde{f}(\mathrm{P}(k), \Gamma)$ converge to $\tilde{f}(\mathrm{P}_{\Gamma}^{*}, \Gamma)$ using the subgradient method with backtracking line search. By doing so, since $\mathrm{P}_{\Gamma}^{*} \to \mathrm{P}^{*}$ and $\tilde{f}(\mathrm{P}_{\Gamma}^{*}, \Gamma) \to v^{*}$ as $k \to \infty$ [14], $\tilde{f}(\mathrm{P}(k), \Gamma)$ will converge to v^{*}.

Theorem 2.1. *Consider an asymmetric network composed of n nodes represented by a weighted digraph. Using Algorithm 2.2, the transmission power vector $P(k)$ asymptotically converges to a stationary vector $P^{*} \in \mathcal{R}$ corresponding to the global minimum of the optimization problem (2.3), v^{*}, as $k \to \infty$.*

Proof. The proof is omitted due to space limitations. ∎

2.4 Simulation Results

Example 2.1. To investigate the efficacy of Algorithm 2.2, consider the experimental asymmetric network in [15] with four nodes, represented by a strongly connected weighted digraph $G = (V, E, \mathbf{W})$. Assume that $\mathcal{P} = [1\ \ 4]^{4}$, i.e. $\mathrm{P}^{\mathrm{low}} = 1_{4}$ and $\mathrm{P}^{\mathrm{up}} = 4 \times 1_{4}$. The initial transmission power vector is chosen as

$$\mathrm{P}^{0} = [2.2\ 1.4\ 1.7\ 2.0]^{\mathrm{T}}, \tag{2.22}$$

which is contained in the compact set $\mathcal{P}$ in this example. Elements of the weight matrix $\mathbf{W}$ of the network are related to the transmission power vector of the network according to (2.2) at every time instant. The resulting initial weight matrix is

$$\mathbf{W} = \begin{bmatrix} 0 & 0.4211 & 0.3462 & 0.2536 \\ 0.4110 & 0 & 0.2706 & 0.3086 \\ 0.1745 & 0.2437 & 0 & 0.2137 \\ 0.3913 & 0 & 0.4417 & 0 \end{bmatrix}.$$

The generalized algebraic connectivity of the directed network corresponding to the initial transmission power vector is $\tilde{\lambda}(\mathrm{P}^0) = 1.0451$. In this example, it is desired to have an algebraic connectivity greater than or equal to $\underline{\lambda} = 0.8$. Hence, the considered initial transmission power vector is strictly feasible, as desired.

To implement Algorithm 2.2, the maximum number of iterations for the inner optimization loop is chosen as $m_{max} = 20$. The design parameters of the backtracking line search are also chosen as $\nu = 0.5, \theta = 0.75,$ and $\alpha_{max} = 1$. Furthermore, the parameter used to numerically calculate the supergradient of the GAC is chosen to be $\beta = 0.01$, and the coefficient by which Γ is multiplied with at the end of an inner optimization loop is at least equal to μ_0 (note that $\mu_0 = 10$). The initial value of Γ is also chosen as $\Gamma^0 = 250$. The performance of the algorithm is evaluated by choosing $\epsilon = 10^{-4}$ in the termination condition of the outer optimization loop. Finally, the search direction is chosen as $v_i(k) = -g_i^-(k)$ for all $i \in \mathbb{N}_n$ and $k \in \mathbb{N}$.

The value of the GAC of the network as the iteration index k increases is shown in Fig. 2.1. The evolution of the transmission power of every node is also demonstrated in Fig. 2.2. The resultant optimal transmission power vector is

$$\mathrm{P}^* = [1.8367\ 1.0064\ 1.6318\ 1.3985]^{\mathrm{T}}, \tag{2.23a}$$

and the corresponding global minimum power and the network GAC, respectively, are

$$v^* = \sum_{i=1}^{4} P_i^* = 5.8734,\ \tilde{\lambda}(\mathrm{P}^*) = 0.8000. \tag{2.23b}$$

The weight matrix associated with the optimal transmission power vector is

$$\mathbf{W}^* = \begin{bmatrix} 0 & 0.3931 & 0.2246 & 0.1676 \\ 0.3483 & 0 & 0.1641 & 0.2358 \\ 0.1598 & 0.2279 & 0 & 0.1970 \\ 0.2156 & 0 & 0.3724 & 0 \end{bmatrix}.$$

In order to verify the obtained results, the *fmincon* function of MATLAB is utilized to solve the optimization problem (2.3) numerically, and proper penalty functions are also incorporated for the violation of the constraints. The output of this function is

$$\mathrm{P}^* = [1.8358\ 1.0002\ 1.6301\ 1.4046]^{\mathrm{T}}, \tag{2.24a}$$

$$v^* = \sum_{i=1}^{n} P_i^* = 5.8707,\ \tilde{\lambda}(\mathrm{P}^*) = 0.8000. \tag{2.24b}$$

Comparing the vectors in (2.23a) and (2.24a), it can be concluded that the obtained

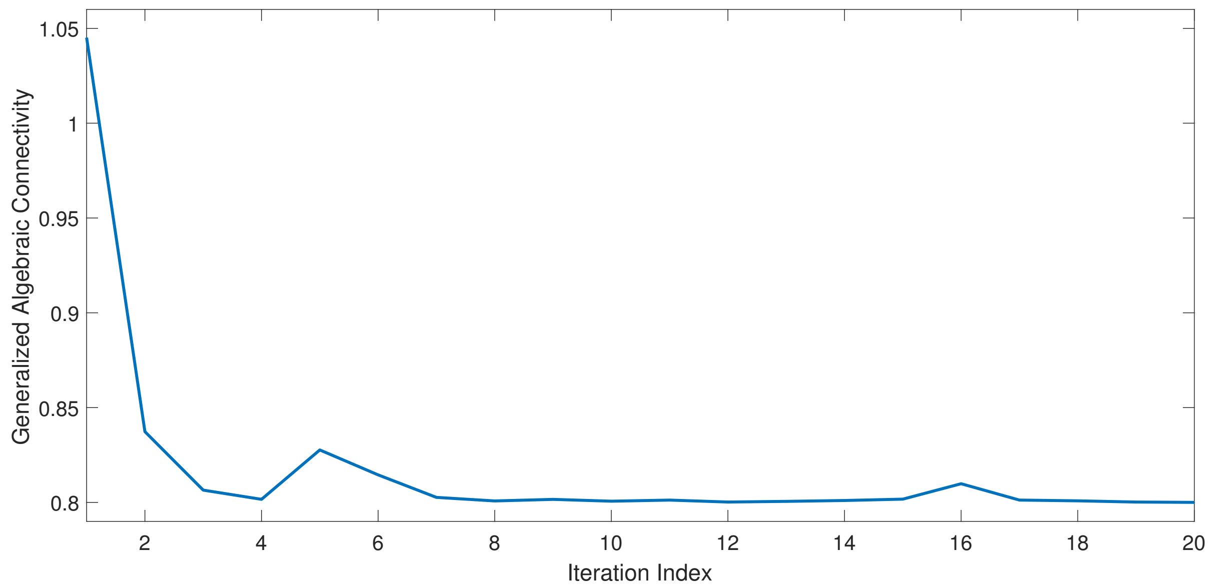

Figure 2.1: Evolution of the generalized algebraic connectivity of the network of Example 2.1.

results are consistent, and in fact the maximum discrepancy between the elements of the two vectors is less than 1%, which is within the numerical error range.

2.5 Conclusion

In this work, the problem of joint transmission power optimization and generalized algebraic connectivity (GAC) control in an asymmetric network represented by a weighted directed graph (digraph) is investigated. The interior point method is utilized to convert the underlying constrained optimization problem into a sequential unconstrained optimization problem. The subgradient method with backtracking line search are then adopted to solve each subproblem numerically. Even though the GAC is a nonconvex,

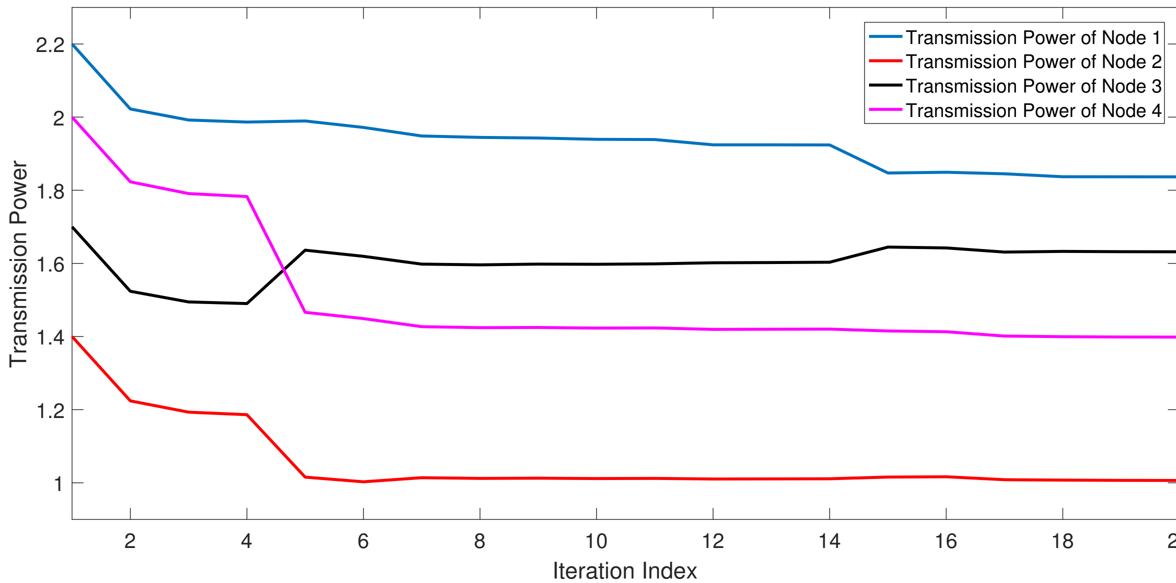

Figure 2.2: Evolution of the transmission power of every node in the network of Example 2.1.

non-smooth continuous function, it is proved that the proposed algorithm converges to the global minimum of the original optimization problem. Efficiency of the algorithm is verified by numerical simulations and comparing the results with MATLAB's *fmincon* function. While the method proposed in this work is centralized, developing the distributed counterpart of the present technique is the main focus of the authors' future work, where the strong connectivity of the network is the key assumption for effective information exchange.

Chapter 3

Lifetime Optimization and Generalized Algebraic Connectivity Control in Asymmetric Networks

In this chapter, the problem of lifetime maximization and connectivity control over asymmetric networks represented by weighted directed graphs (digraphs) is investigated using a centralized approach. Each node is assumed to deplete its battery linearly with respect to the transmission powers used for communicating with its neighbors. Lifetime of the network is defined as the minimum lifetime over all nodes and is formulated as a function of these transmission power levels. The notion of generalized algebraic connectivity (GAC), used as the network connectivity measure, is also formulated as an implicit function of the network's transmission power matrix. An optimization problem is presented to maximize the network lifetime while satisfying constraints on the GAC and transmission power. The mixed interior point-exterior point method is utilized to transform the constrained optimization problem into a sequential unconstrained problem. Each subproblem is solved numerically using the subgradient method with backtracking line search. Asymptotic convergence of the proposed algorithm to the global optimum of the original optimization problem is demonstrated analytically. The effectiveness of the algorithm is verified by simulations.

This chapter is based on the following publication:

M. Esmaeilpour, A. G. Aghdam, and S. Blouin, "Lifetime optimization and generalized algebraic connectivity control in asymmetric networks," submitted for conference publication.

Unlike Chapters 2 and 4 where the interior point method is used to numerically handle the optimization problems, the mixed interior point-exterior point method is used in this chapter as a result of considering equality constraints in the optimization problem. Ultimately, note that as mentioned before, the concatenation of all transmission powers in Chapters 3 and 4 is a matrix, whereas it is a vector in Chapter 2.

3.1 Introduction

Sensor networks consist of geographically distributed autonomous sensor nodes, which have sensing, processing and communication capabilities [17]. These networks have applications in environmental monitoring, surveillance, target localization, etc. [22]. The communication links between the sensors may either be bi- or uni-directional, resulting in symmetric and asymmetric networks, respectively. An example of the former is the wireless sensor networks (WSN), whereas underwater acoustic sensor networks (UWASN) are a type of asymmetric networks [28], where bi-directional communication links may not be possible due to several sources of uncertainty [15].

The key issue in deploying sensor networks is the power consumption of the sensor nodes, which directly impacts the lifetime of the network [19]. The sensor nodes are typically battery-powered, and since replacing the battery of a dead node may not be cost-effective nor straightforward, especially in the case of UWASNs, careful battery energy management is crucial. The network lifetime is typically defined as the time until the first sensor node depletes its energy [17, 21]. The importance of maximizing the network lifetime has motivated numerous studies in the literature, where it is considered as an explicit performance metric [17]. For instance, the authors of [21] consider a routing problem in static wireless ad hoc networks, where the objective of maximizing the network lifetime is solved via a shortest path routing algorithm. In [19], an optimal control approach is used to solve the problem of routing in sensor networks with the goal of maximizing the network's lifetime. The authors consider a dynamic energy consumption model for the batteries. In a fixed topology, they show that there exists an optimal policy consisting of time-invariant routing probabilities. The authors extend these results further in [17] where they consider a more general state space battery model. A mobile sensor network for monitoring a moving target is investigated in [22]. The authors propose a technique determining a near-optimal relocation strategy for the sensors and an energy-efficient route for transferring information from the target to the destination. In [23], the authors propose base station mobility to counter the suboptimal energy dissipation of some nodes in a WSN. To prolong the network lifetime, various mobility patterns are considered and using a mixed integer programming framework, their impact on the network lifetime is characterized. The authors of [24] provide a mathematical model for network lifetime maximization integrating WSN design decisions on sensor places, activity schedules, data routes and trajectory of the mobile sink(s). They then present two heuristic approaches for the solution of the model and show its efficacy via numerical experiments. For further algorithms on network lifetime maximization, the interested reader is referred to survey studies similar to [25] and the references therein.

In addition to lifetime, another important issue to address in the deployment of sensor networks is their connectivity. Networks run distributed algorithms to determine a (often-global) quantity from local measurements. It is well-known that the convergence rate of these cooperative algorithms is directly related to the connectivity level of the network [28]. Some of the papers in the literature that consider connectivity in the context of prolonging the network lifetime can be found in [25]. Among the different connectivity measures, algebraic connectivity may be used as a measure for symmetric networks, where its counterpart for asymmetric networks is the generalized algebraic connectivity (GAC) [28]. Even though it is desirable to have a highly connected network, it would mean significantly higher total power consumption. Given the inverse relationship between the network lifetime and the total power consumption, an unreasonably high connectivity level in the network would deplete the nodes' batteries prematurely. Hence, an appropriate balance between the lifetime of the network and its connectivity level is imperative.

The optimization problem considered in this chapter aims to maximize the lifetime of the network while satisfying certain constraints on the GAC and the transmission powers and ensuring that the nodes run out of energy at the same time. In our earlier work [28], a similar problem was considered; however, the main objective in [28] is minimizing the total transmission power, which has been shown to rapidly deplete energy from some nodes, ultimately reducing the overall network lifetime [17]. This is in contrast with the goal of this study. To the best of our knowledge, there is currently no study in the literature that considers the GAC in the same context as network lifetime maximization. The methodologies of this study and [28] are also different, as the method of [28] cannot handle the eqaulity constraints considered here. unlike [28], it is assumed here that nodes use different transmission powers to communicate with their neighbors. The mixed interior point-exterior point method of [13] is utilized to convert the resulting nonlinear constrainted optimization problem to a sequential unconstrained optimization problem. The subgradient method with backtracking line search is then used to solve each subproblem numerically, and asymptotic convergence of the proposed algorithm to the global optimum of the original problem is demonstrated analytically. Efficacy of the proposed method is shown by numerical simulations, where an experimental UWASN is considered as an example of an asymmetric network. The distributed version of the algorithm proposed in this study is submitted to a journal and is currently under review [30].

The remainder of the chapter is organized as follows. In Section II, first, notations and preliminary graph theory concepts used throughout the chapter are given, followed by the optimization problem and its numerical solution. In Section III, convergence analysis of the proposed algorithm is presented. The simulation results are subsequently provided in Section IV, and finally, Section V contains the concluding remarks.

3.2 Prelimineries

Notation: Throughout this chapter, the set of real numbers greater than r is denoted by $\mathbb{R}_{>r}$, and the finite set of natural numbers $\{1, 2, ..., n\}$ is denoted by $\mathbb{N}_n$. The superscript T is used to indicate the transpose of a real vector or matrix. Moreover, the inner product of two real matrices $\mathbf{v},\mathbf{w} \in \mathbb{R}^{n\times n}$ is represented by $\langle \mathbf{v}, \mathbf{w} \rangle$, and the function $\mathrm{tr}(\cdot)$ denotes the trace of a given real matrix. The real part of a complex number $c \in \mathbb{C}$ is denoted by $\Re(c)$, and $\lceil . \rceil$ denotes the ceiling function. $\mathbb{B}_\sigma(\cdot)$ is a closed ball of radius $\sigma \in \mathbb{R}_{>0}$ centered around the considered point.

For any $k \in \mathbb{N}$, let $G(k) = (V, E(k), \mathbf{W}(k))$ denote a weighted directed graph (digraph) in the time interval $[t_k, t_{k+1})$, characterized by a set of vertices $V = \mathbb{N}_n$, a set of edges $E(k)$, and a weight matrix $\mathbf{W}(k) \in \mathbb{R}^{n\times n}$. The (i, j) element of the weight matrix, denoted by $w_{ij}(k)$, is the weight associated with the communication link from node j to node i, $\vec{ji} \in E(k)$, in the time interval $[t_k, t_{k+1})$ for any pair of distinct nodes $i, j \in \mathbb{N}_n$ and any $k \in \mathbb{N}$. Furthermore, the out-neighbor and in-neighbor sets associated with node i and the Laplacian of the weighted digraph $G(k)$ are denoted by $N_i^{out}(k)$, $N_i^{in}(k)$ and $\mathbf{L}(k) \in \mathbb{R}^{n\times n}$, respectively. The generalized algebraic connectivity (GAC) of a weighted digraph $G(k)$ is defined as the smallest real part of the nonzero eigenvalues of the Laplacian matrix $\mathbf{L}(k)$ and is denoted by $\tilde{\lambda}(\mathbf{L}(k))$ for any $k \in \mathbb{N}$ [4]. As shown in [4], the imaginary part of the eigenvalue corresponding to the GAC is not related to the convergence rate of the distributed algorithms running over the network. Additionally, if a network has multiple connected components, but is not strongly connected as a whole, the GAC would only be applicable to each component separately.

3.2.1 Problem Formulation

Consider a time-varying asymmetric network with n stationary nodes, whose information exchange topology is represented by the weighted digraph $G(k)$ for all $k \in \mathbb{N}$, as noted earlier. As a more general case, unlike the authors' earlier work [28], it is assumed that each node uses different power levels to communicate with their neighbors. Therefore, the transmission power matrix is denoted by $\mathbf{P}(k) = [P_{ji}(k)] \in \mathbb{R}^{n\times n}$, for $i, j \in \mathbb{N}_n$ and $k \in \mathbb{N}$, where $P_{ji}(k) \in [P_{ji}^{\text{low}}, P_{ji}^{\text{up}}]$ is the transmission power that node i uses to transmit information to node j. P_{ji}^{low} and P_{ji}^{up} are, respectively, the fixed lower bound and upper bound of the permissible transmission power $P_{ij}(k)$, which are known *a priori*. In a noise-limited environment, the relation between the transmission power $P_{ji}(k)$ and the weight of the corresponding communication link $w_{ji}(k)$ can be described by a function of the following form

$$w_{ji}(k) = h(P_{ji}(k); \xi_{ji}), \tag{3.1}$$

for any $i, j \in \mathbb{N}_n$, where ξ_{ji} represents a set of real constant parameters characterizing the communication channel $\vec{ij}$, and $h(.;.)$ is an increasing continuously differentiable function [1].

Expression (3.1) enables the formulation of the network GAC as an implicit function of the power matrix, i.e., $\tilde{\lambda}(\mathbf{P}(k))$ for any $k \in \mathbb{N}$. Unlike the algebraic connectivity, introduced in [27] for undirected networks, which is concave in its domain [31], the

GAC is neither concave nor convex, and hence, is referred to as a nonconvex function. Furthermore, the GAC is piecewise differentiable over finite intervals, but can be non-differentiable in certain points. The discontinuity in the derivative of $\tilde{\lambda}(\mathbf{P}(k))$ occurs when the eigenvalue of the Laplacian matrix corresponding to the GAC changes from a real eigenvalue to a complex conjugate pair or vice versa [2].

Given the transmission power vector $\mathrm{P}_i(k) = [P_{1i}(k), \cdots, P_{ni}(k)]^T$ for any $i \in \mathbb{N}_n$ and $k \in \mathbb{N}$, the lifetime of node i is given by

$$T(\mathrm{P}_i(k)) = \frac{\mathfrak{e}_i^0}{\sum_{j \in N_i^{out}} q_{ji} \cdot P_{ji}(k) \cdot \tau_{ji} \cdot K_{ji} + \sum_{j \in N_i^{in}} q_{ij} \cdot \mathfrak{e}_{ij}^r \cdot K_{ij}} \tag{3.2}$$

[21], where q_{ji} denotes the transmission rate per unit of time from node i to node j (like bits per second), τ_{ji} shows the time that node i has to keep transmitting one packet of information with transmission power P_{ji} to ensure it has been received at node j, and K_{ji} is the number of information packets sent from node i to node j. Moreover, $\mathfrak{e}_i^0$ is the initial energy of node i battery, and $\mathfrak{e}_{ij}^r$ is the energy used by node i to receive information from node j. Note that $T(\mathrm{P}_i(k))$, for all $i \in \mathbb{N}_n$, is a concave function on its domain and has a unit of time. The energy required to sense incoming information packets is considered to be negligible compared to transmission power requirements [17]. The denominator of (3.2) is relatively simple. However, the underlying optimization problem and the proposed algorithm are not critically dependent on the exact energy consumption model. The system (or network) lifetime is accordingly defined as the

minimum lifetime over all nodes [17, 21], i.e., for all $i \in \mathbb{N}_n$ and any $k \in \mathbb{N}$, the network lifetime is as

$$T_{sys}(\mathbf{P}(k)) = \min_{i \in \mathbb{N}_n} T(\mathrm{P}_i(k)). \tag{3.3}$$

In a noise-limited environment, a higher transmission power leads to a more connected network [28]. However, as evident from (3.2), it will lead to a decrease in the network lifetime. Since an appropriate balance between the network lifetime and its connectivity level is imperative for the efficient operation of the network, the following optimization problem is considered for the network

$$\begin{aligned} \underset{\mathbf{P}}{\text{minimize}} \quad & -T_{sys}(\mathbf{P}) \\ \text{subject to} \quad & \tilde{\lambda}(\mathbf{P}(k)) \geq \underline{\lambda}, \ \forall k \in \mathbb{N}, \\ & P_{ji}^{\text{low}} \leq P_{ji}(k) \leq P_{ji}^{\text{up}}, \ \forall i, j \in \mathbb{N}_n, \forall k \in \mathbb{N}, \\ & T(\mathrm{P}_i(k)) = T_{sys}(\mathbf{P}(k)), \ \forall i \in \mathbb{N}_n, \forall k \in \mathbb{N}, \end{aligned} \tag{3.4}$$

where $\underline{\lambda}$ is a prespecified constant, reflecting the lowest acceptable connectivity level. It has been shown that maximizing the network lifetime may result in the simultaneous depletion of the node energies [17, 19, 22]. To this end, the lifetime constraints are considered in (3.4) to ensure that nodes deplete their energies at the same time. This is also favorable from a practical point of view because if the entire network was to deplete its energy at once, battery replacement can be performed more efficiently.

From (3.4), for all $i, j \in \mathbb{N}_n$ and $k \in \mathbb{N}$, define

$$f(\mathbf{P}(k)) = -T_{sys}(\mathbf{P}(k))$$

as the main cost function, and

$$h^1(k) = \tilde{\lambda}(\mathbf{P}(k)) - \underline{\lambda},$$

$$h^2_{ji}(k) = P^{\text{up}}_{ji} - P_{ji}(k),$$

$$h^3_{ji}(k) = P_{ji}(k) - P^{\text{low}}_{ji},$$

$$h^4_i(k) = T(\mathrm{P}_i(k)) - T_{sys}(\mathbf{P}(k)),$$

as the constraint functions. To solve the optimization problem (3.4), the mixed interior point-exterior point method of [13] is utilized, which transforms a constrained optimization problem into a sequential unconstrained problem. To this end, it is noted that when $k \to \infty$, problem (3.4) has the same minimizers as the following problem [13]

$$\underset{\mathbf{P}}{\text{minimize}} \quad \tilde{f}(\mathbf{P}, \Gamma, \Upsilon), \tag{3.5}$$

where

$$\tilde{f}(\mathbf{P}(k), \Gamma, \Upsilon) = f(\mathbf{P}(k)) + \Gamma^{-1} \cdot I(\mathbf{P}(k)) + \Upsilon \cdot O(\mathbf{P}(k)), \tag{3.6}$$

$$I(\mathbf{P}(k)) = -\log\left(h^1(k)\right) - \sum_{i\in\mathbb{N}_n}\sum_{j\in\mathbb{N}_n} \log\left(h^2_{ji}(k)\right) - \sum_{i\in\mathbb{N}_n}\sum_{j\in\mathbb{N}_n} \log\left(h^3_{ji}(k)\right), \tag{3.7}$$

Algorithm 3.1. Mixed interior point-exterior point method

1: Given $\mathbf{P} = \mathbf{P}^0 \in \mathbb{R}^{n\times n}$, initialize
   $\Gamma = \Gamma^0$ and $\Upsilon = \Upsilon^0 \in \mathbb{R}_{>0}$.
2: Choose arbitrary constants $\mu, \delta \in \mathbb{R}_{>1}$ and $\epsilon \in \mathbb{R}_{>0}$.
3: **while** $m\Gamma^{-1} > \epsilon$
4:   Compute $\mathbf{P}^* \in \underset{\mathbf{P}\in\mathcal{R}}{\operatorname{argmin}}\ \tilde{f}(\mathbf{P}, \Gamma, \Upsilon)$.
5:   $\mathbf{P} = \mathbf{P}^*$
6:   $\Gamma = \mu\Gamma$
7:   $\Upsilon = \delta\Upsilon$
8: **end while**

$$O(\mathbf{P}(k)) = \sum_{i\in\mathbb{N}_n} h_i^4(k)^2, \tag{3.8}$$

and $\Gamma, \Upsilon \in \mathbb{R}_{>0}$ are updated as the optimization process moves forward. Algorithm 3.1 shows the procedure to numerically solve the optimization problem (3.5). The set $\mathcal{R}$ in the algorithm is defined as the domain of the logarithmic barrier function (3.7). The initial transmission power matrix $\mathbf{P}^0$ needs to be strictly inside the feasible set $\mathcal{R}$. There are five parameters ϵ, μ, δ, Γ^0, and Υ^0 in Algorithm 3.1 that need to be chosen appropriately. The choice of ϵ involves a trade-off between the accuracy and the execution speed of the algorithm. The parameters μ and δ determine the rate of increase of Γ and Υ, respectively. The parameters Γ^0 and Υ^0 determine the initial weights given to the penalty functions $I(\mathbf{P}(k))$ and $O(\mathbf{P}(k))$, respectively.

Since (3.5) is a sequential optimization problem, in order to solve each subproblem (line 4 of Algorithm 3.1), the subgradient method is utilized. Note that since the GAC is a nonconvex function, the joint cost function will also be a nonconvex function. To this end, the definition of a matrix-valued function's subgradient is given next. Note that a supergradient of a nonconvex function has the same definition as in Definition 3.1, except the inequality is flipped. The superdifferential set is also defined similarly to the subdifferential set.

Definition 3.1. Matrix $\mathbf{g}$ is said to be the subgradient of the nonconvex and non-differentiable function $\tilde{f} : \mathbb{R}^{n \times n} \to \mathbb{R}$ at $\mathbf{P} \in \mathcal{R}$ if there exists a real scalar $\sigma \in \mathbb{R}_{>0}$ such that for any $\hat{\mathbf{P}} \in \mathbb{B}_\sigma(\mathbf{P})$, the following inequality holds:

$$\tilde{f}(\hat{\mathbf{P}}, \Gamma, \Upsilon) \geq \tilde{f}(\mathbf{P}, \Gamma, \Upsilon) + \langle \hat{\mathbf{P}} - \mathbf{P}, \mathbf{g} \rangle. \tag{3.9}$$

The set of all subgradients of function $\tilde{f}$ at $\mathbf{P} \in \mathcal{R}$ is called the subdifferential set $\partial \tilde{f}(\mathbf{P}, \Gamma, \Upsilon)$ [2].

The subgradient method moves the current iteration of the optimization loop in the opposite direction of a subgradient of the function to be optimized ($\tilde{f}(\mathbf{P}(k), \Gamma, \Upsilon)$ in this case). That is, at any time t_k, $k \in \mathbb{N}$,

$$\mathbf{P}(k+1) = \mathbf{P}(k) + \alpha(k) \cdot \mathbf{v}(k), \tag{3.10}$$

where $\mathbf{v}(k) = -\mathbf{g}(k)$ is the search direction in the time interval $[t_k, t_{k+1})$, and $\mathbf{g}(k) = [g_{ij}(k)] \in \partial \tilde{f}(\mathbf{P}(k), \Gamma, \Upsilon)$, for $i, j \in \mathbb{N}_n$, is an arbitrary subgradient of the cost function $\tilde{f}$. Also, for any $k \in \mathbb{N}$, $\alpha(k)$ is the step-size, determining the magnitude of the move along the search direction. At any time t_k, $k \in \mathbb{N}$, and for any $i, j \in \mathbb{N}_n$, the $(i, j)^{\text{th}}$

element of $\mathbf{g}(k)$ is given by

$$g_{ij}(k) = -\frac{\partial T_{sys}(\mathbf{P}(k))}{\partial P_{ij}(k)} + \Gamma^{-1}\frac{\partial I(\mathbf{P}(k))}{\partial P_{ij}(k)} + \Upsilon\frac{\partial O(\mathbf{P}(k))}{\partial P_{ij}(k)} \tag{3.11}$$

where

$$\frac{\partial I(\mathbf{P}(k))}{\partial P_{ij}(k)} = -\frac{\nabla\tilde{\lambda}_{ij}(\mathbf{P}(k))}{\tilde{\lambda}(\mathbf{P}(k)) - \underline{\lambda}} + \frac{1}{P_{ji}^{\mathrm{up}} - P_{ij}(k)} - \frac{1}{P_{ij}(k) - P_{ji}^{\mathrm{low}}}, \tag{3.12}$$

$$\frac{\partial O(\mathbf{P}(k))}{\partial P_{ij}(k)} = 2\sum_{l\in\mathbb{N}_n}\big(\frac{\partial T_l(\mathrm{P}_l(k))}{\partial P_{ij}(k)} - \frac{\partial T_{sys}(\mathbf{P}(k))}{\partial P_{ij}(k)}\big) \times \big(T_l(\mathrm{P}_l(k)) - T_{sys}(\mathbf{P}(k))\big). \tag{3.13}$$

In (3.12), $\nabla\tilde{\lambda}_{ij}(\mathbf{P}(k))$ is the subgradient of the GAC with respect to $P_{ij}(k)$, and can be arbitrarily chosen from the subdifferential set

$$\partial\tilde{\lambda}_{ij}(\mathbf{P}(k)) \coloneqq \frac{1}{\beta(k)}\{\partial\tilde{\lambda}^+_{ij}(\mathbf{P}(k)), \partial\tilde{\lambda}^-_{ij}(\mathbf{P}(k))\}$$

where $\beta(k) = \beta_0/k^{\beta_0}$, for some $\beta_0 \in \mathbb{R}_{>0}$, is the step-size used to numerically compute $\partial\tilde{\lambda}^+_{ij}(\mathbf{P}(k))$ and $\partial\tilde{\lambda}^-_{ij}(\mathbf{P}(k))$ given by [2, 28]

$$\partial\tilde{\lambda}^+_{ij}(\mathbf{P}(k)) = \tilde{\lambda}(\mathbf{P}(k) + \beta(k)\mathbf{e}_{ij}) - \tilde{\lambda}(\mathbf{P}(k)),$$

$$\partial\tilde{\lambda}^-_{ij}(\mathbf{P}(k)) = \tilde{\lambda}(\mathbf{P}(k)) - \tilde{\lambda}(\mathbf{P}(k) - \beta(k)\mathbf{e}_{ij}).$$

Note that $\beta(k)$ was assumed to be a real constant in our earlier work [28], whereas it is time-varying in this study. Let also $\mathbf{g}^+(k)$ and $\mathbf{g}^-(k)$, respectively, correspond to the cases where $\nabla\tilde{\lambda}_{ij}(\mathbf{P}(k)) = \frac{1}{\beta(k)}\partial\tilde{\lambda}^+_{ij}(\mathbf{P}(k))$ and $\nabla\tilde{\lambda}_{ij}(\mathbf{P}(k)) = \frac{1}{\beta(k)}\partial\tilde{\lambda}^-_{ij}(\mathbf{P}(k))$, for all

$i, j \in \mathbb{N}_n$. The partial derivatives of the nodes' lifetimes are easily found via (3.2). To compute the partial derivatives of the network lifetime for each $k \in \mathbb{N}$, one first needs to determine which node the minimum lifetime corresponds to before using (3.2).

The procedure proposed for computing a solution of the sequential unconstrained optimization problem (3.5) is presented in Algorithm 3.2. The parameter m_{max} is the maximum number of iterations the inner optimization loop is repeated. In lines 12 to 15 of Algorithm 3.2, the step-size $\alpha(k)$ is computed via the backtracking line search method, providing the maximum allowable step length to move along the obtained search direction. This line search method starts with a large estimate of the step-size α_{max} and iteratively reduces the step-size by the factor $\theta \in (0, 1)$ until a sufficient decrease determined by the parameter $\nu \in (0, 1)$ is observed in the cost function. Unlike the gradient method, the subgradient approach does not result in a descent direction at every time instant t_k, $k \in \mathbb{N}$. For the case where $\mathbf{v}(k)$ is not a descent direction (as specified later in Remark 3.1), the multiplicative factors by which Γ and Υ are increased (μ and δ, respectively) are updated first, such that they satisfy the inequality given later in Lemma 3.1, and then the inner optimization loop ends (lines 7 to 11 of Algorithm 3.2). This increases the likelihood of having a descent search direction in the next optimization iteration. The parameters μ_0 and δ_0 in Algorithm 3.2 are the initial values of the parameters μ and δ, respectively.

Remark 3.1. For the non-differentiable function $\tilde{f}(\mathbf{P}(k), \Gamma, \Upsilon)$ with subdifferential set

$\partial\tilde{f}(\mathbf{P}(k), \Gamma, \Upsilon)$ at $\mathbf{P}(k) \in \mathcal{R}$, $\mathbf{v}(k)$ is not a descent direction if

$$\exists\ \mathbf{g}(k) \in \partial\tilde{f}(\mathbf{P}(k), \Gamma, \Upsilon) \text{ such that } \langle \mathbf{v}(k), \mathbf{g}(k) \rangle \geq 0, \tag{3.15}$$

for any $k \in \mathbb{N}$ and $\Gamma, \Upsilon \in \mathbb{R}_{>0}$.

Note that Remark 3.1 is the same result as the Lemma 1 of [28]. The only notable difference is that the inner product in (3.15) is between two matrices, compared to the inner product of two vectors in [28]. This is the result of considering a transmission power matrix in this study, rather than a transmission power vector.

Remark 3.2. As can be understood from Definition 3.1, Remark 3.1 is just a sufficient condition for $\mathbf{v}(k)$ to be a descent direction. Thus, even if $\langle \mathbf{v}(k), \mathbf{g}(k) \rangle < 0$ for a $k \in \mathbb{N}$, $\mathbf{v}(k)$ may still not be a descent direction. In such a case, since the cost function will not decrease sufficiently when moving along the search direction, the resulting step-size from the backtracking algorithm will be almost zero (the parameter ζ in line 16 of Algorithm 3.2 determines the threshold below which the step-size can be considered approximately zero). To this end, by terminating the inner optimization loop and updating the values of μ and δ according to line 17 of Algorithm 3.2, the likelihood of having a descent search direction in the next iteration increases (see Remark 3.4).

Finally, to solve the optimization problems described by (3.4) and (3.5), Assumption 3.1 is assumed to hold. Unlike [28], edge set of the digraph representing the network may change as the optimization algorithm proceeds.

Assumption 3.1. *The network digraph is assumed to be strongly connected at all times, meaning that there is a directed path from every node in the graph to every other node.*

Regarding the scalability and run-time of the proposed algorithm, the most computationally heavy component of the algorithm is the calculation of the GAC of the network. Whichever algorithm or software is utilized to determine the GAC, the GAC needs to be calculated at most $2(n^2 - n) + 1$ times at each iteration of the inner optimization loop. The run-time of the rest of the elements of Algorithm 3.2, such as the backtracking algorithm, and the convergence rate of the interior point method are not dependent on the network size.

3.3 Convergence Analysis of the Optimization Algorithm

The asymptotic convergence of the proposed sequantial Algorithm 3.2 to the global optimum of the constrained optimization (3.4) is provided in this section.

Remark 3.3. Consider an asymmetric network composed of n nodes represented by a weighted digraph, as described earlier, and let Assumption 3.1 hold. Using the mixed interior point-exterior point algorithm, the following relations hold

1. $\lim_{k\to\infty} f(\mathbf{P}(k)) = v^*$,
2. $\lim_{k\to\infty} \tilde{f}(\mathbf{P}(k), \Gamma, \Upsilon) = v^*$,

Algorithm 3.2. Sequential unconstrained optimization

Given $\mathbf{P} = \mathbf{P}^0 \in \mathbb{R}^{n \times n}$, initialize
$\Gamma = \Gamma^0$, $\Upsilon = \Upsilon^0 \in \mathbb{R}_{>0}$ and $k = 1$.
Choose arbitrary constants $\nu \in (0,1)$, $\theta \in (0,1)$,
$\mu_0, \delta_0 \in \mathbb{R}_{>1}$, β_0, ϵ, $\alpha_{max} \in \mathbb{R}_{>0}$, $m_{max} \in \mathbb{N}$ and consider
the prescribed parameters $\underline{\lambda} \in \mathbb{R}_{>0}$, and $\mathbf{P}^{\text{up}}$, $\mathbf{P}^{\text{low}} \in \mathbb{R}^{n \times n}$.
while $m\Gamma^{-1} > \epsilon$ **do**
$\mu = \mu_0$, $\delta = \delta_0$
  for $m_{iter} = 1 : m_{max}$ **do**
    Compute $\mathbf{v}(k)$
    Compute $Q_{ij}(k), \frac{\partial I^1(\mathbf{P}(k))}{\partial P_{ij}(k)}, \frac{\partial I^2(\mathbf{P}(k))}{\partial P_{ij}(k)}$ for $i, j \in \mathbb{N}_n$
    according to Lemma 3.1.
    if $\exists \mathbf{g}(k) \in \partial \tilde{f}(\mathbf{P}(k), \Gamma, \Upsilon)$ **such that** $\langle \mathbf{v}(k), \mathbf{g}(k) \rangle \geq 0$ **do**
      $\mu = \max\{\mu_0, \lceil \frac{\sum_{i \in \mathbb{N}_n} \sum_{j \in \mathbb{N}_n} Q_{ij}(k)(\frac{\partial I^1(\mathbf{P}(k))}{\partial P_{ij}(k)} + \frac{\partial I^2(\mathbf{P}(k))}{\partial P_{ij}(k)})}{\Gamma \sum_{i \in \mathbb{N}_n} \sum_{j \in \mathbb{N}_n} Q_{ij}(k)^2} - 1 \rceil\}$
      $\delta = 1$
      break
    $\alpha(k) = \alpha_{max}$
    while $\mathbf{P}(k) + \alpha(k) \cdot \mathbf{v}(k) \notin \mathcal{R}$ **or**
  $\tilde{f}(\mathbf{P}(k) + \alpha(k) \cdot \mathbf{v}_k, \Gamma, \Upsilon) > \tilde{f}(\mathbf{P}(k), \Gamma, \Upsilon) + \nu \cdot \alpha(k) \cdot \langle \mathbf{v}(k), \mathbf{g}(k) \rangle$ **do**
      $\alpha(k) = \theta \cdot \alpha(k)$
    end while
    if $\alpha(k) \leq \zeta$ **do**
      $\mu = \mu_0$, $\delta = 1$
      break
    $\mathbf{P}(k+1) = \mathbf{P}(k) + \alpha(k) \cdot \mathbf{v}(k)$
    $k = k + 1$
  end
  $\Gamma = \mu\Gamma$
  $\Upsilon = \delta\Upsilon$
end while

3. $\lim_{k\to\infty} \Gamma^{-1} \cdot I(\mathbf{P}(k)) = 0$,

4. $\lim_{k\to\infty} \Upsilon \cdot O(\mathbf{P}(k)) = 0$,

5. $\lim_{k\to\infty} \mathbf{P}(k) = \mathbf{P}^*$,

where v^* is the global optimum of the optimization problem (3.4), and $\mathbf{P}^*$ is its corresponding minimizer [13, 28].

Remark 3.3 states that there exists a sequence $\mathbf{P}(k)$ for optimization problem (3.5) that converges to the global minimum of the problem (3.4). Now one needs to show that the subgradient method with backtracking line search can generate such a sequence. To this end, the following lemma and theorem are presented.

Assume that for some $k \in \mathbb{N}$, $\mathbf{v}(k)$ is not a descent direction according to either Remark 3.1 or 3.2. Based on Algorithm 3.2, the optimization algorithm ends at iteration k without updating the transmission power matrix, and the values of Γ and Υ are increased accordingly. Similar to [28], the iteration index k does not increase, but rather the optimization loop in the time interval $[t_k, t_{k+1})$ is repeated. For the sake of convergence analysis, this new iteration index is denoted by k'.

Lemma 3.1. *If $\boldsymbol{v}(k)$ is not a descent direction for some $k \in \mathbb{N}$ according to Remark 3.1, it will be a descent direction at iteration k', if*

$$\mu \geq \lceil \frac{\sum_{i\in\mathbb{N}_n} \sum_{j\in\mathbb{N}_n} -Q_{ij}(k)(\frac{\partial I^1(\boldsymbol{P}(k))}{\partial P_{ij}(k)} + \frac{\partial I^2(\boldsymbol{P}(k))}{\partial P_{ij}(k)})}{\Gamma \sum_{i\in\mathbb{N}_n} \sum_{j\in\mathbb{N}_n} Q_{ij}(k)^2} - 1 \rceil, \tag{3.16}$$

and $\delta = 1$, *where*

$$Q_{ij}(k) = -\frac{\partial T_{sys}(\boldsymbol{P}(k))}{\partial P_{ij}(k)} + \Upsilon \frac{\partial O(\boldsymbol{P}(k))}{\partial P_{ij}(k)}, \tag{3.17}$$

and $\frac{\partial I^1(\boldsymbol{P}(k))}{\partial P_{ij}(k)}$ *and* $\frac{\partial I^2(\boldsymbol{P}(k))}{\partial P_{ij}(k)}$ *are the partial derivatives of the function* $I(\boldsymbol{P}(k))$ *given by (3.7), where the superscripts correspond to the (possibly) different supergradients of the GAC used in obtaining these values.*

Proof. According to Remark 3.1, if $\exists \mathbf{g}(k) \in \partial \tilde{f}(\mathbf{P}(k), \Gamma, \Upsilon)$ such that

$$\langle \mathbf{v}(k), \mathbf{g}(k) \rangle = \sum_{i \in \mathbb{N}_n} \sum_{j \in \mathbb{N}_n} \Big(-(-\frac{\partial T_{sys}(\mathbf{P}(k))}{\partial P_{ij}(k)} + \Gamma^{-1} \frac{\partial I^1(\mathbf{P}(k))}{\partial P_{ij}(k)} + \Upsilon \frac{\partial O(\mathbf{P}(k))}{\partial P_{ij}(k)}) \times$$
$$(-\frac{\partial T_{sys}(\mathbf{P}(k))}{\partial P_{ij}(k)} + \Gamma^{-1} \frac{\partial I^2(\mathbf{P}(k))}{\partial P_{ij}(k)} + \Upsilon \frac{\partial O(\mathbf{P}(k))}{\partial P_{ij}(k)}) \Big) \geq 0, \tag{3.18}$$

then $\mathbf{v}(k)$ is not a descent direction. Given the definition of $Q_{ij}(k)$ in (4.17), by simplifying (4.28), one has

$$\langle \mathbf{v}(k), \mathbf{g}(k) \rangle = \sum_{i \in \mathbb{N}_n} \sum_{j \in \mathbb{N}_n} \Big(-Q_{ij}(k)^2 - \Gamma^{-1} Q_{ij}(k) \big(\frac{\partial I^1(\mathbf{P}(k))}{\partial P_{ij}(k)} + \frac{\partial I^2(\mathbf{P}(k))}{\partial P_{ij}(k)} \big) -$$
$$\Gamma^{-2} \frac{\partial I^1(\mathbf{P}(k))}{\partial P_{ij}(k)} \cdot \frac{\partial I^2(\mathbf{P}(k))}{\partial P_{ij}(k)} \Big) \geq 0. \tag{3.19}$$

For iteration k', by considering $\delta = 1$, we will have $Q_{ij}(k') = Q_{ij}(k)$. Furthermore, since the inner optimization loop is terminated without updating the transmission power matrix, we obtain

$$\langle \mathbf{v}(k'), \mathbf{g}(k') \rangle = \sum_{i\in\mathbb{N}_n}\sum_{j\in\mathbb{N}_n} \Big(- Q_{ij}(k)^2 - \mu^{-1}\Gamma^{-1} Q_{ij}(k) \big(\frac{\partial I^1(\mathbf{P}(k))}{\partial P_{ij}(k)} + \frac{\partial I^2(\mathbf{P}(k))}{\partial P_{ij}(k)} \big) - \mu^{-2}\Gamma^{-2} \frac{\partial I^1(\mathbf{P}(k))}{\partial P_{ij}(k)} \cdot \frac{\partial I^2(\mathbf{P}(k))}{\partial P_{ij}(k)} \Big). \quad (3.20)$$

It is possible to make (3.20) less than zero by appropriately choosing the value of μ. To this end, the right-hand side of (3.19) is multiplied with $\frac{1}{\mu^2}$ and after manipulations, it becomes

$$\sum_{i\in\mathbb{N}_n}\sum_{j\in\mathbb{N}_n} \Big(- Q_{ij}(k)^2 - \frac{1}{\mu\Gamma} Q_{ij}(k) \big(\frac{\partial I^1(\mathbf{P}(k))}{\partial P_{ij}(k)} + \frac{\partial I^2(\mathbf{P}(k))}{\partial P_{ij}(k)} \big) - \frac{1}{\mu^2\Gamma^2} \frac{\partial I^1(\mathbf{P}(k))}{\partial P_{ij}(k)} \cdot \frac{\partial I^2(\mathbf{P}(k))}{\partial P_{ij}(k)} \Big) \leq \sum_{i\in\mathbb{N}_n}\sum_{j\in\mathbb{N}_n} \Big(- \frac{\mu^2+1}{\mu^2} Q_{ij}(k)^2 - \frac{\mu+1}{\mu^2\Gamma} Q_{ij}(k) \big(\frac{\partial I^1(\mathbf{P}(k))}{\partial P_{ij}(k)} + \frac{\partial I^2(\mathbf{P}(k))}{\partial P_{ij}(k)} \big) - \frac{2}{\mu^2\Gamma^2} \frac{\partial I^1(\mathbf{P}(k))}{\partial P_{ij}(k)} \cdot \frac{\partial I^2(\mathbf{P}(k))}{\partial P_{ij}(k)} \Big). \quad (3.21)$$

It is evident that the left-hand side of (3.21) is equal to (3.20). Now, if the right-hand side of (3.21) is less than zero, it is guaranteed that $\langle \mathbf{v}(k'), \mathbf{g}(k') \rangle < 0$. By substituting (3.19) in the right-hand side of (3.21), it is easy to show that if the following inequality holds,

$$\mu \geq \Big\lceil \frac{\sum_{i\in\mathbb{N}_n}\sum_{j\in\mathbb{N}_n} -Q_{ij}(k) \big(\frac{\partial I^1(\mathbf{P}(k))}{\partial P_{ij}(k)} + \frac{\partial I^2(\mathbf{P}(k))}{\partial P_{ij}(k)} \big)}{\Gamma \sum_{i\in\mathbb{N}_n}\sum_{j\in\mathbb{N}_n} Q_{ij}(k)^2} - 1 \Big\rceil,$$

one will have $\langle \mathbf{v}(k'), \mathbf{g}(k') \rangle < 0$. This completes the proof. ■

Remark 3.4. If $\mathbf{v}(k)$, for some $k \in \mathbb{N}$, is not a descent direction according to Remark 3.2, a logical way to obtain a descent direction at iteration k' is to determine how μ and δ should be updated in order to have

$$\tilde{f}(\mathbf{P}(k) + \alpha(k')\mathbf{v}(k'), \Gamma', \Upsilon') - \tilde{f}(\mathbf{P}(k), \Gamma', \Upsilon') < 0,$$

for some $\alpha(k') \in (0, \alpha_{max}]$. However, the nonlinearity and complexity of the function $\tilde{f}$ is a barrier to achieving this. As a remedy, one can update the values of μ and δ based on line 17 of Algorithm 3.2. It is easy to see that this will make the value of $\langle \mathbf{v}(k'), \mathbf{g}(k') \rangle$ more negative, increasing the likelihood of having a descent direction at iteration k'.

For each Γ and Υ, denote the central point of the interior point-exterior point by $\mathbf{P}^*_{\Gamma,\Upsilon}$, where $\tilde{f}(\mathrm{P}^*_{\Gamma,\Upsilon}, \Gamma, \Upsilon) \to v^*$ as $k \to \infty$ [14].

Theorem 3.1. *Consider an asymmetric network composed of n nodes represented by a weighted digraph, as described earlier. Using the interior point method in conjunction with the subgradient approach and the backtracking line search to solve the constrained optimization problem (3.4), the transmission power matrix $\boldsymbol{P}(k)$ asymptotically converges to a stationary matrix $\boldsymbol{P}^* \in \mathcal{R}$ corresponding to the global minimum of the optimization problem (3.4), v^*, as $k \to \infty$.*

Proof. Given $\mathbf{g}(k) \in \partial\tilde{f}(\mathbf{P}(k), \Gamma, \Upsilon)$, for any $k \in \mathbb{N}$ and $\Gamma, \Upsilon \in \mathbb{R}_{>0}$, the search direction $\mathbf{v}(k)$ is a descent direction as per the previously discussed lemmas and remarks. As a result, the backtracking line search algorithm will eventually stop for some $\alpha(k) \in$

$(0, \alpha_{max}]$. Then

$$\tilde{f}(\mathbf{P}(k), \Gamma, \Upsilon) - \tilde{f}(\mathbf{P}(k-1), \Gamma, \Upsilon) \leq \nu \cdot \alpha(k-1) \cdot \langle \mathbf{v}(k-1), \mathbf{g}(k-1) \rangle. \tag{3.22}$$

Given that $\langle \mathbf{v}(k), \mathbf{g}(k) \rangle < 0$, for any $k \in \mathbb{N}$, (3.22) can be rewritten as

$$\tilde{f}(\mathbf{P}(k), \Gamma, \Upsilon) - \tilde{f}(\mathbf{P}(k-1), \Gamma, \Upsilon) \leq 0. \tag{3.23}$$

From the definition of subgradient (Definition 3.1), any $\mathbf{g}(k) \in \partial \tilde{f}_l(\mathbf{P}(k), \Gamma)$, including the one corresponding to the search direction ($\mathbf{v}(k) = -\mathbf{g}(k)$), satisfies the following inequality

$$\tilde{f}(\mathbf{P}(k-1), \Gamma, \Upsilon) + \langle \mathbf{P}(k-1) - \mathbf{P}^*_{\Gamma,\Upsilon}, \mathbf{g}(k-1) \rangle \leq \tilde{f}(\mathbf{P}^*_{\Gamma,\Upsilon}, \Gamma, \Upsilon). \tag{3.24}$$

Combining (3.23) and (3.24) yields

$$\tilde{f}(\mathbf{P}(k), \Gamma, \Upsilon) - \tilde{f}(\mathbf{P}^*_{\Gamma,\Upsilon}, \Gamma, \Upsilon) \leq \langle \mathbf{P}(k-1) - \mathbf{P}^*_{\Gamma,\Upsilon}, \mathbf{g}(k-1) \rangle. \tag{3.25}$$

The right-hand side of (3.25) can then be rewritten as

$$\frac{1}{\alpha(k-1)} \langle \mathbf{P}(k-1) - \mathbf{P}^*_{\Gamma,\Upsilon}, \alpha(k-1)\mathbf{g}(k-1) \rangle = \frac{1}{\alpha(k-1)} \langle \mathbf{P}(k-1) - \mathbf{P}^*_{\Gamma,\Upsilon}, \mathbf{P}(k) - \mathbf{P}(k-1) \rangle. \tag{3.26}$$

By manipulating the second term in the inner product of the right-hand side of (3.26), (3.25) becomes

$$\tilde{f}(\mathbf{P}(k), \Gamma, \Upsilon) - \tilde{f}(\mathbf{P}^*_{\Gamma,\Upsilon}, \Gamma, \Upsilon) \leq \frac{-1}{\alpha(k-1)} \langle \mathbf{P}(k-1) - \mathbf{P}^*_{\Gamma,\Upsilon}, \mathbf{P}(k-1) - \mathbf{P}^*_{\Gamma,\Upsilon} \rangle$$
$$+ \frac{1}{\alpha(k-1)} \langle \mathbf{P}(k-1) - \mathbf{P}^*_{\Gamma,\Upsilon}, \mathbf{P}(k) - \mathbf{P}^*_{\Gamma,\Upsilon} \rangle, \quad (3.27)$$

and after further manipulations, one arrives at

$$\tilde{f}(\mathbf{P}(k), \Gamma, \Upsilon) - \tilde{f}(\mathbf{P}^*_{\Gamma,\Upsilon}, \Gamma, \Upsilon) \leq \frac{-1}{\alpha(k-1)} \langle \mathbf{P}(k-1) - \mathbf{P}^*_{\Gamma,\Upsilon}, \mathbf{P}(k-1) - \mathbf{P}^*_{\Gamma,\Upsilon} \rangle +$$
$$\frac{1}{\alpha(k-1)} \langle \mathbf{P}(k) - \mathbf{P}^*_{\Gamma,\Upsilon}, \mathbf{P}(k) - \mathbf{P}^*_{\Gamma,\Upsilon} \rangle + \frac{1}{\alpha(k-1)} \langle \mathbf{P}(k-1) - \mathbf{P}(k), \mathbf{P}(k) - \mathbf{P}^*_{\Gamma,\Upsilon} \rangle.$$
$$(3.28)$$

The third term in the right-hand side of the above inequality is rewritten to obtain

$$\tilde{f}(\mathbf{P}(k), \Gamma, \Upsilon) - \tilde{f}(\mathbf{P}^*_{\Gamma,\Upsilon}, \Gamma, \Upsilon) \leq \frac{-1}{\alpha(k-1)} \langle \mathbf{P}(k-1) - \mathbf{P}^*_{\Gamma,\Upsilon}, \mathbf{P}(k-1) - \mathbf{P}^*_{\Gamma,\Upsilon} \rangle +$$
$$\frac{1}{\alpha(k-1)} \langle \mathbf{P}(k) - \mathbf{P}^*_{\Gamma,\Upsilon}, \mathbf{P}(k) - \mathbf{P}^*_{\Gamma,\Upsilon} \rangle + \langle \mathbf{g}(k-1), \mathbf{P}^*_{\Gamma,\Upsilon} - \mathbf{P}(k) \rangle. \quad (3.29)$$

The right-hand side of (3.25) is re-expressed as

$$\tilde{f}(\mathbf{P}^*_{\Gamma,\Upsilon}, \Gamma, \Upsilon) - \tilde{f}(\mathbf{P}(k-1), \Gamma, \Upsilon) \geq \langle \mathbf{g}(k-1), \mathbf{P}^*_{\Gamma,\Upsilon} - \mathbf{P}(k) \rangle + \alpha(k) \langle \mathbf{g}(k-1), \mathbf{g}(k-1) \rangle.$$
$$(3.30)$$

Now, combining (3.29) and (3.30) yields

$$\tilde{f}(\mathbf{P}(k),\Gamma,\Upsilon)+\tilde{f}(\mathbf{P}(k-1),\Gamma,\Upsilon)-2\tilde{f}(\mathbf{P}^*_{\Gamma,\Upsilon},\Gamma,\Upsilon)\leq$$
$$\frac{-1}{\alpha(k-1)}\langle\mathbf{P}(k-1)-\mathbf{P}^*_{\Gamma,\Upsilon},\mathbf{P}(k-1)-\mathbf{P}^*_{\Gamma,\Upsilon}\rangle+\frac{1}{\alpha(k-1)}\langle\mathbf{P}(k)-\mathbf{P}^*_{\Gamma,\Upsilon},\mathbf{P}(k)-\mathbf{P}^*_{\Gamma,\Upsilon}\rangle-$$
$$\alpha(k)\langle\mathbf{g}(k-1),\mathbf{g}(k-1)\rangle. \quad (3.31)$$

Note that $\langle\mathbf{g}(k-1),\mathbf{g}(k-1)\rangle$ in the right-hand side of (3.31) is non-negative. In addition, $\tilde{f}(\mathbf{P}(k-1),\Gamma,\Upsilon)-\tilde{f}(\mathbf{P}^*_{\Gamma,\Upsilon},\Gamma,\Upsilon)\geq 0$ for any $\Gamma,\Upsilon\in\mathbb{R}_{>0}$. Hence, (3.31) can be rewritten as

$$\tilde{f}(\mathbf{P}(k),\Gamma,\Upsilon)-\tilde{f}(\mathbf{P}^*_{\Gamma,\Upsilon},\Gamma,\Upsilon)\leq\frac{1}{\alpha(k-1)}\langle\mathbf{P}(k)-\mathbf{P}^*_{\Gamma,\Upsilon},\mathbf{P}(k)-\mathbf{P}^*_{\Gamma,\Upsilon}\rangle$$
$$-\frac{1}{\alpha(k-1)}\langle\mathbf{P}(k-1)-\mathbf{P}^*_{\Gamma,\Upsilon},\mathbf{P}(k-1)-\mathbf{P}^*_{\Gamma,\Upsilon}\rangle. \quad (3.32)$$

Averaging both sides of (3.32) yields

$$\frac{1}{k}\sum_{j=2}^{k}\left(\tilde{f}(\mathbf{P}(j),\Gamma,\Upsilon)-\tilde{f}(\mathbf{P}^*_{\Gamma,\Upsilon},\Gamma,\Upsilon)\right)\leq\frac{1}{k}\sum_{j=2}^{k}\Big(\frac{1}{\alpha(j-1)}\langle\mathbf{P}(j)-\mathbf{P}^*_{\Gamma,\Upsilon},\mathbf{P}(j)-\mathbf{P}^*_{\Gamma,\Upsilon}\rangle$$
$$-\frac{1}{\alpha(j-1)}\langle\mathbf{P}(j-1)-\mathbf{P}^*_{\Gamma,\Upsilon},\mathbf{P}(j-1)-\mathbf{P}^*_{\Gamma,\Upsilon}\rangle\Big), \quad (3.33)$$

which can be simplified to

$$\frac{1}{k}\sum_{j=2}^{k}\big(\tilde{f}(\mathbf{P}(j),\Gamma,\Upsilon)-\tilde{f}(\mathbf{P}^*_{\Gamma,\Upsilon},\Gamma,\Upsilon)\big)\leq$$
$$\frac{1}{k}\cdot\frac{1}{\min\alpha(j-1)|_{j=2}^{k}}\cdot\Big(\langle\mathbf{P}(k)-\mathbf{P}^*_{\Gamma,\Upsilon},\mathbf{P}(k)-\mathbf{P}^*_{\Gamma,\Upsilon}\rangle-\langle\mathbf{P}(1)-\mathbf{P}^*_{\Gamma,\Upsilon},\mathbf{P}(1)-\mathbf{P}^*_{\Gamma,\Upsilon}\rangle\Big). \tag{3.34}$$

Note that both the inner products in the right-hand side of (3.34) are non-negative. Also, given the definition of the central points $\mathbf{P}^*_{\Gamma,\Upsilon}$, one has

$$\tilde{f}(\mathbf{P}(k),\Gamma,\Upsilon)-\tilde{f}(\mathbf{P}^*_{\Gamma,\Upsilon},\Gamma,\Upsilon)\leq\frac{1}{k}\sum_{j=2}^{k}\big(\tilde{f}(\mathbf{P}(j),\Gamma,\Upsilon)-\tilde{f}(\mathbf{P}^*_{\Gamma,\Upsilon},\Gamma,\Upsilon)\big). \tag{3.35}$$

Now, (3.34) is transformed to

$$\tilde{f}(\mathbf{P}(k),\Gamma,\Upsilon)-\tilde{f}(\mathbf{P}^*_{\Gamma,\Upsilon},\Gamma,\Upsilon)\leq\frac{1}{k}\cdot\frac{1}{\min\alpha(j-1)|_{j=2}^{k}}\cdot\langle\mathbf{P}(k)-\mathbf{P}^*_{\Gamma,\Upsilon},\mathbf{P}(k)-\mathbf{P}^*_{\Gamma,\Upsilon}\rangle. \tag{3.36}$$

According to Algorithm 3.2, every time the inner optimization loop is terminated, the values of Γ and Υ increase and $\tilde{f}(\mathbf{P}^*_{\Gamma,\Upsilon},\Gamma,\Upsilon)$ and $\mathbf{P}^*_{\Gamma,\Upsilon}$ are updated. As $k\to\infty$, the right-hand side of (3.36) approaches zero, and also $\tilde{f}(\mathrm{P}^*_{\Gamma,\Upsilon},\Gamma,\Upsilon)\to v^*$. This means that as $k\to\infty$, $\tilde{f}(\mathbf{P}(k),\Gamma,\Upsilon)$ converges to v^*. Also, according to Remark 3.3, $\mathbf{P}(k)$ approaches $\mathbf{P}^*$ as k increases. This completes the proof. ∎

3.4 Simulation Results

Example 3.1. To investigate the efficacy of Algorithm 3.2, consider the four-node experimental asymmetric network of [15]. First, it is assumed that the communication link from node 2 to node 4 is not feasible in this network due to environmental constraints, i.e., $P_{42}(k)$ is equal to zero for all $k \in \mathbb{N}$ and is not considered in the optimization algorithm. The rest of the edge set, however, may change. Assuming that for every $i, j \in \mathbb{N}_n$, $P_{ij}^{\text{low}} = 0$ and $P_{ij}^{\text{up}} = 4$, the initial transmission power matrix is chosen as

$$\mathbf{P}^0 = \begin{bmatrix} 0 & 1.3 & 1.5 & 1.6 \\ 1.2 & 0 & 1.5 & 1.3 \\ 1.7 & 1.7 & 0 & 1.4 \\ 1.5 & 0 & 1.5 & 0 \end{bmatrix}.$$

Using (3.1), the resulting initial weight matrix of the corresponding digraph is as

$$\mathbf{W} = \begin{bmatrix} 0 & 0.3411 & 0.1227 & 0.1163 \\ 0.3813 & 0 & 0.2966 & 0.2914 \\ 0.1745 & 0.2437 & 0 & 0.1420 \\ 0.2461 & 0 & 0.3857 & 0 \end{bmatrix},$$

where the network's initial GAC is $\tilde{\lambda}(\mathbf{P}^0) = 0.7148$. Assuming

$$\mathbf{q} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}, \ \mathbf{K} = \begin{bmatrix} 0 & 2 & 1 & 2 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 1 \\ 1 & 0 & 2 & 0 \end{bmatrix}, \ \tau = \begin{bmatrix} 0 & 1.2 & 1.5 & 2 \\ 1.1 & 0 & 1.8 & 1.4 \\ 1.5 & 1.7 & 0 & 0.8 \\ 1.2 & 0 & 0.9 & 0 \end{bmatrix},$$

$\mathfrak{e}_i^0 = 20$ and $\mathfrak{e}_{ij}^r = 1.5$ for every $i, j \in \mathbb{N}_n$, the initial lifetime of nodes are

$$T(\mathbf{P}^0) = [1.2723, 1.6653, 1.4652, 1.2771]^T.$$

Furthermore, it is desired to have a GAC greater than or equal to $\underline{\lambda} = 0.7$. Hence, the considered initial transmission power matrix is strictly inside the feasible set $\mathcal{R}$.

To implement Algorithm 3.2, the following previously explained design parameters are considered: $m_{max} = 29$, $\nu = 0.01$, $\theta = 0.72$, $\alpha_{max} = 1$, $\beta_0 = 0.01$, $\mu_0 = 5$, $\delta_0 = 1.2$, $\Gamma^0 = 2$, $\Upsilon^0 = 1$ and $\epsilon = \zeta = 10^{-6}$. The search direction is chosen as $\mathbf{v}(k) = -\mathbf{g}^{-}(k)$ for all $k \in \mathbb{N}$.

Fig. 3.1 shows the individual node lifetimes as the iteration index k increases, where the minimum of all at each instant is the network lifetime (not shown in the figure). The evolution of the various transmission powers of nodes is presented in Fig.

3.2. The resultant optimal transmission power matrix is

$$\mathbf{P}^* = \begin{bmatrix} 0 & 0.7717 & 1.7398 & 0.3975 \\ 0.7272 & 0 & 0.8460 & 0.9917 \\ 0.0003 & 1.5554 & 0 & 2.0393 \\ 1.8316 & 0 & 0.2034 & 0 \end{bmatrix},$$

resulting in the following weight matrix

$$\mathbf{W}^* = \begin{bmatrix} 0 & 0.2682 & 0.1935 & 0.0002 \\ 0.2903 & 0 & 0.1209 & 0.2328 \\ 0 & 0.2102 & 0 & 0.2967 \\ 0.3440 & 0 & 0.0986 & 0 \end{bmatrix}.$$

As it can be seen from the above optimal matrices and Fig. 3.2, certain communication links have been removed when compared with the initial edge set of the network. In general, the transmission powers used for communication between nodes have decreased, which is understandable given the inverse relationship between the lifetime and transmission power defined in (3.2); however, for certain communication links, the powers have increased, which is counter-intuitive. This can be attributed to the asymmetric nature of the considered network. The corresponding network GAC is $\tilde{\lambda}(\mathbf{P}^*) = 0.7000$.

The obtained optimal lifetimes are as

$$T(\mathbf{P}^*) = [1.9050, 1.9054, 1.9050, 1.9051]^T,$$

indicating an increase of almost 50% compared to the initial network lifetime. Also, note that all the nodes now have similar lifetimes as desired.

To compare these results with another method, the *fmincon* function of MATLAB is utilized to solve the optimization problem (3.4) with the same initial conditions, and appropriate penalty functions are considered for the violation of the constraints. The output of this function is obtained as

$$\mathbf{P}^* = \begin{bmatrix} 0 & 0.7410 & 2.6549 & 0.0054 \\ 0 & 0 & 0.0777 & 2.2314 \\ 0.0778 & 1.7693 & 0 & 0.0210 \\ 2.5439 & 0 & 0.3688 & 0 \end{bmatrix},$$

which leads to a network GAC of $\tilde{\lambda}(\mathbf{P}^*) = 0.7485$ and the following lifetimes

$$T(\mathbf{P}^*) = [1.8542, 1.8542, 1.8542, 1.8542]^T.$$

Since the user has the freedom of choosing the design parameters in the proposed algorithm, the resulting optimal solution is slightly better in comparison to that of the

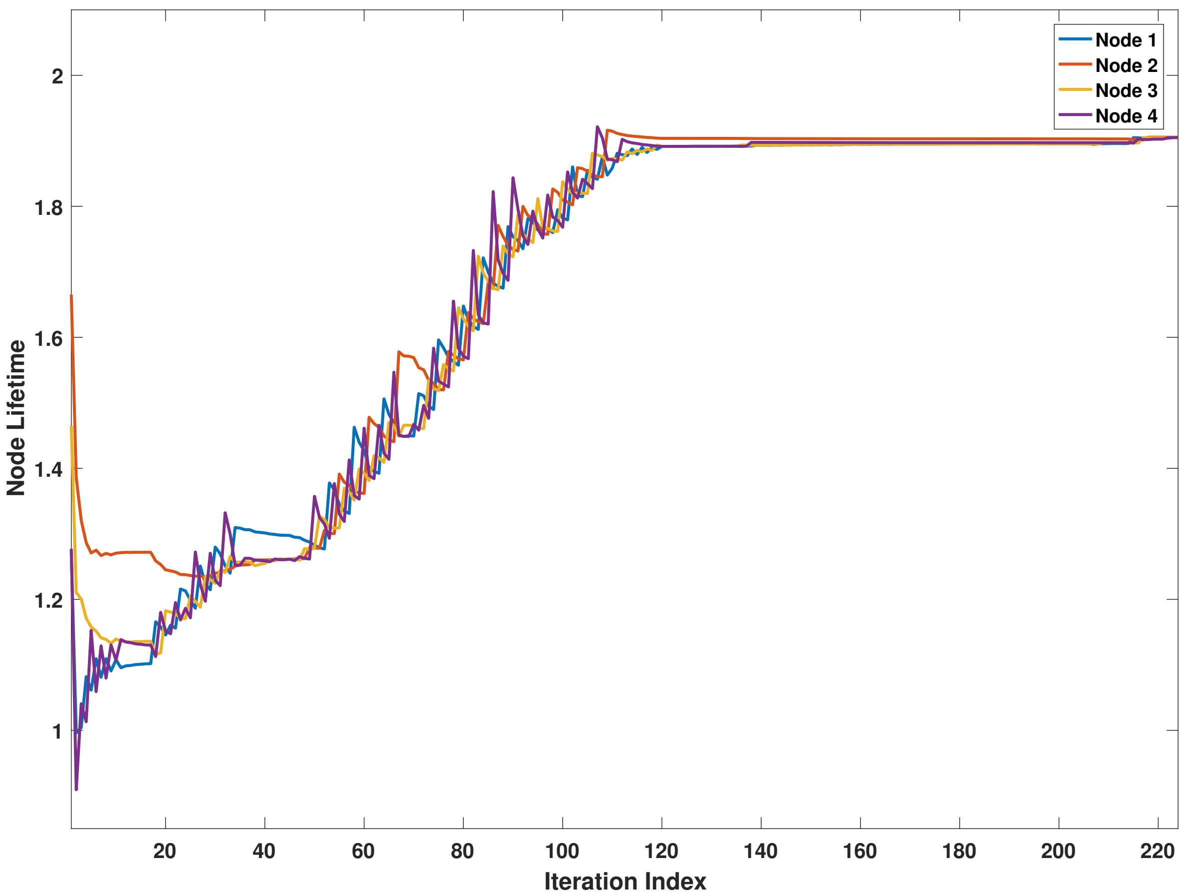

Figure 3.1: Evolution of the lifetimes of the nodes of the network of Example 3.1.

fmincon function. The output of the *fmincon* function has converged to the neighborhood of the same optimal point obtained via the centralized approach; however, it has stopped prematurely. It can also be seen that the two methods have different minimizers for the same optimal point.

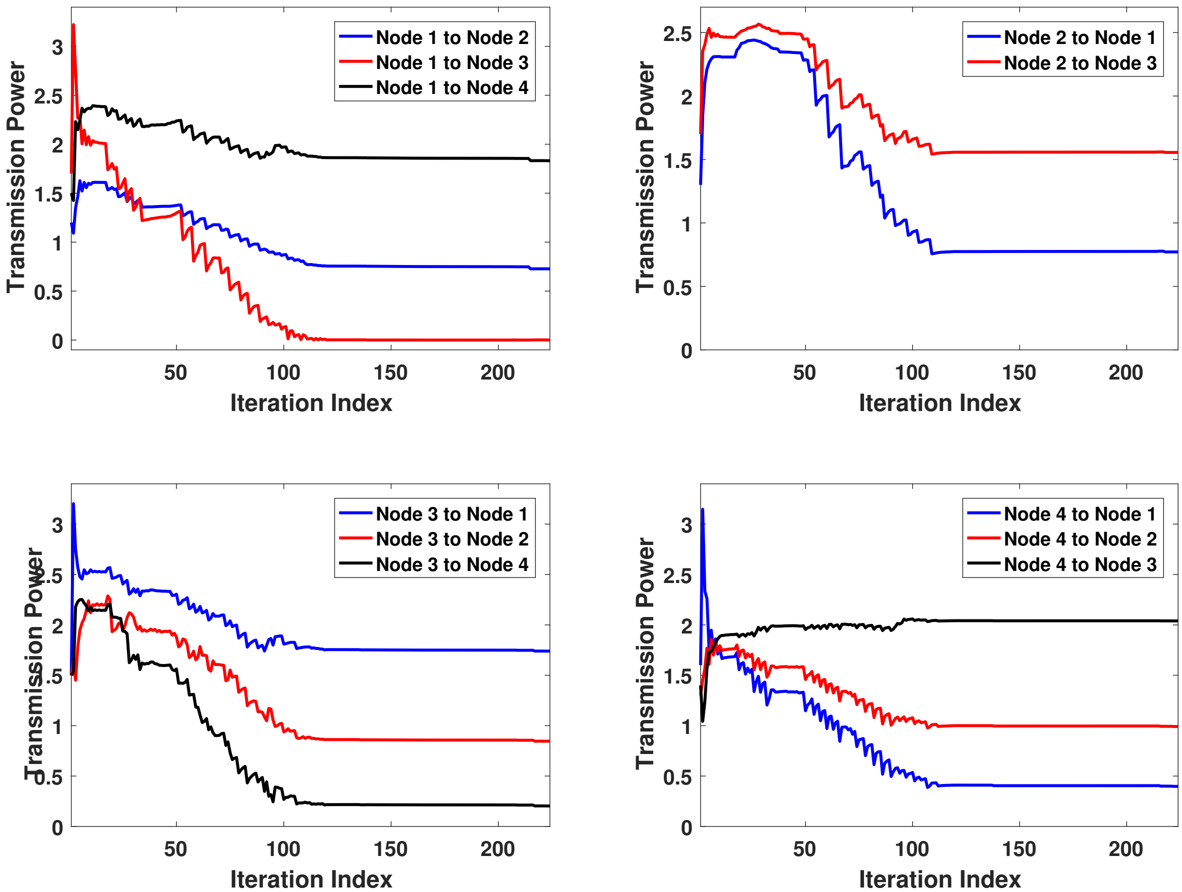

Figure 3.2: Evolution of the transmission power for nodes making up the network of Example 3.1.

3.5 Conclusion

In this work, the problem of joint network lifetime maximization and generalized algebraic connectivity (GAC) control in an asymmetric network, represented by a weighted directed graph, is investigated in a joint manner. The mixed interior point-exterior point method is employed to convert the resulting constrained optimization problem into a sequential unconstrained problem. The subgradient method with backtracking line search are then adopted to solve each subproblem. Unlike the gradient method, the subgradient approach may not necessarily be a descent direction at each optimization iteration. This issue is addressed as well. It is proved accordingly that the proposed method converges asymptotically to the global optimum of the optimization problem. Efficacy of the algorithm is verified by numerical simulations, and the results obtained are compared with another method. The obtained optimal transmission power matrices lead to the simultaneous depletion of the node energies. Furthermore, the results obtained for some nodes are counter-intuitive, as they suggest increasing certain transmission power levels to achieve better network lifetime.

Chapter 4

Connectivity, Transmission Power, and Lifetime Optimization in Asymmetric Networks: A Distributed Approach

In this chapter, three problems over asymmetric networks represented by weighted directed graphs (digraphs) are investigated using a distributed approach. The first problem relates to transmission power control over the network to maximize connectivity. It is assumed that different nodes use different transmission power levels to communicate with their neighbors. The notion of generalized algebraic connectivity (GAC), used as a network connectivity measure, is formulated as an implicit function of the network's transmission power matrix. An optimization problem is introduced to maximize the network GAC while satisfying constraints on communication transmission powers. The second problem is the dual of the first one, i.e., minimizing the total transmission power of the network while controlling the network GAC. Ultimately, an optimization problem is formulated to maximize the lifetime of the network and control its connectivity. Each node is assumed to deplete its battery linearly with respect to the transmission powers used for communication. The network lifetime is defined as the minimum lifetime over all nodes and is formulated as a function of the transmission power levels used. The interior point method is utilized to transform the mentioned constrained optimization problems into sequential unconstrained optimization problems. Each subproblem is then solved numerically via the subgradient method with backtracking line search. Asymptotic convergence of the proposed algorithms to a local or global optima of the original optimization problems are demonstrated analytically. The effectiveness of the proposed distributed algorithm is verified by simulations.

This chapter is based on the following journal paper submission:

M. Esmaeilpour, A. G. Aghdam, and S. Blouin, "Connectivity, transmission power, and lifetime optimization in asymmetric networks: A distributed approach," submitted for journal publication.

4.1 Introduction

Ad-hoc networks are composed of spatially distributed fixed or mobile sensors capable of sensing, processing and exchanging data without the need for a pre-existing framework. The challenges involved in deploying effective networks and the recent advances in computation, communication and sensing have stimulated substantial research in this area of study [4, 26, 28]. There is a plethora of applications using these networks, such as environmental monitoring, target detection and localization, disaster control, smart farming, etc. [15,18,25,26]. A graph can be used to represent the communication links of the deployed network, which may either be symmetric or asymmetric. An example of a symmetric network is the terrestrial wireless sensor networks (WSN), and an example of an asymmetric network, where the communication link between two distinct nodes are often uni-directional, is underwater acoustic sensor networks (UWASN) [4,28,55]. In the latter example, several sources of uncertainty and noise contribute to this asymmetric nature, which include, but are not limited to, multipath propagation, inconsistencies in the shape of the seabed, sound speed profile variations, temperature fluctuations, and nearby shipping activities [15,36,55]. Another difference between the terrestrial and underwater sensor networks is that contrary to the WSNs which may consist of hundreds of nodes for a speific application, the number of deployed nodes in UWASNs is much smaller. For instance, the experimental network of [15] consists of only four nodes.

Several issues need to be addressed in deploying sensor networks, and in particular, the connectivity of the network and its lifetime are two of the most important issues. Sensor networks typically utilize cooperative algorithms in order to determine specific (often-global) quantities using only local information. It is well-known that the convergence rate of these algorithms is directly related to the network connectivity, and that a highly connected network diffuses information more efficiently [4, 28, 29]. Additionally, having an algebraic connectivity measure allows one to apply mathematical tools such as differential operators on the considered measure. Algebraic connectivity has been used as a measure of connectivity in symmetric networks, and is defined as the second smallest eigenvalue of the Laplacian matrix of the (weighted) undirected graph representing the network [27]. There are many studies in the literature investigating algebraic connectivity in symmetric networks. For instance, a distributed algorithm is presented in [3] to estimate and control the algebraic connectivity of symmetric networks using a stochastic power iteration method. In [37], a distributed method, which relies on the distributed computation of the powers of the adjacency matrix, is proposed to obtain both upper and lower bounds at each iteration for the algebraic connectivity of a symmetric network. As the algorithm proceeds, these bounds converge to the true value of the algebraic connectivity. In addition, a supergradient algorithm is used along with a decentralized eigenvector estimation strategy in [5] to maximize the algebraic connectivity of a symmetric network.

Unlike symmetric networks, the equivalent of algebraic connectivity in asymmetric networks has not been investigated as much. A simple extension of algebraic connectivity to directed graphs is proposed in [38], where the magnitude of the smallest nonzero eigenvalue of the Laplacian matrix is presented as a measure of connectivity. This notion, however, fails to capture any operational characteristic of the network [4]. To address this shortcoming, the notion of generalized algebraic connectivity (GAC) is introduced in [39] as the real part of the smallest nonzero eigenvalue of the Laplacian matrix of the weighted directed graph (digraph) representing the network, and is shown to be directly related to the asymptotic convergence rate of continuous-time consensus algorithms running over the network. An algorithm based on the subspace consensus approach is proposed in [4] for distributed computation of the GAC using only local information. Furthermore, in [28], the GAC is formulated as a function of the transmission power vector of the network, and then a distributed supergradient algorithm is proposed to maximize the GAC.

In addition to connectivity, another critical aspect of a network is the power consumption of the nodes, directly affecting its lifetime. Power consumption in sensor networks is either communication-related or non-communication-related. In some applications, such as a UWASN, it is the communication-related part that plays a dominant role in power consumption [17, 21]. Sensor nodes are typically battery-powered, and recharging or replacing their batteries is not always a viable option. Incapacitation of some nodes due to battery depletion can result in a disconnected network, which in turn can prevent the network from completing its mission [20, 23]. Therefore, an appropriate power management scheme is essential for the efficient operation of any sensor network. Typically, network lifetime is defined as the time it takes for the first node to completely deplete its energy [17, 21, 22]. There are numerous studies in the literature where, due to the significance of a network's lifetime, it is considered as an explicit performance metric. In [17], the authors study the problem of maximizing the network lifetime via routing, where they consider a general state-space battery model for the nodes. They show that in a fixed topology, there exists an optimal policy consisting of time-invariant routing probabilities. They also consider a joint routing and initial energy allocation problem, and prove that the optimal policy depletes the energy reserves of all nodes simultaneously. In [23], base station mobility is proposed as a remedy for countering inefficient routing and topology in WSNs. The authors build a framework to characterize the impact of various mobility patterns on the network lifetime and conclude that optimal Gaussian and spiral patterns result in the highest lifetime values. A mobile sensor network for monitoring a moving target is investigated in [22], where an algorithm is developed to find a near-optimal relocation strategy for the sensors as well as an energy-efficient route for transferring information from the target to destination. The author of [40] proposes an optimal distance-based transmission strategy based on ant colony optimization to maximize the lifetime of WSNs and demonstrate the effectiveness of their findings by simulations. In [41], the joint optimal design of the physical, medium access control, and network layers is considered to maximize the lifetime of WSNs with limited available energy. The optimization problem is formulated by taking into account several network variables such as the routing flow, transmission rate, etc. The Gauss-Seidel algorithm, in conjunction with the gradient method, is subsequently used to update the considered network variables. For further studies on network lifetime maximization, the interested reader is referred to recent survey studies such as [25] and [26].

In a noise-limited environment, higher transmission powers used by the nodes for communication result in better and stronger communication links, which normally means that the network will be more connected [28, 36]. Even though having a highly connected network is desirable, it may require higher total transmission power. On the other hand, given the inverse relationship between the network lifetime and its power consumption, higher total transmission power would lead to shorter network lifetime. Given the importance of the connectivity and lifetime of the network as discussed previously, it is imperative to determine an appropriate balance between the two. To this end, three optimization problems are considered in this chapter over asymmetric networks represented by weighted digraphs. In the first problem, the objective is to maximize the GAC of the network while satisfying constraints on the total transmission power of the network and the individual transmission power values. To the best of our knowledge, [2] is the only paper in the literature that aims at maximizing connectivity in asymmetric networks but it does not consider any limit on the total power consumption of the network. In the second problem, it is desired to minimize the total transmission power of the network while ensuring that connectivity is maintained above a certain level, and that the power values are bounded within prescribed limits. This work extends the results of [28] by considering a transmission power matrix for a more general formulation, and also, by proposing a distributed approach to solve the underlying optimization problem using local information. In [1], the total transmission power is minimized while a constraint on some non-algebraic connectivity measure is imposed; the constraint in the present chapter, however, is imposed on the GAC. The third optimization problem relates to network lifetime maximization subject to constraints on the GAC and transmission power values. In our earlier work [29], a similar problem is considered; however, this study extends the results of [29] by proposing a distributed optimization algorithm. The reason for considering both the second and the third problems is that it has been shown in the literature that the second optimization problem can reduce the overall network lifetime [17]. We aim to determine the extent to which this result holds in the context of strongly-connected asymmetric networks with constraints on the connectivity level and the values of transmission powers. The interior point method is used to transform the resultant constrained optimization problems into unconstrained ones. The subgradient method along with a novel approximate backtracking line search is utilized to solve the above subproblems. The case where the subgradient method does not yield a descent direction is examined in detail. Convergence of the proposed distributed algorithm to a local or global optima of the optimization problems is shown analytically, and its efficacy is demonstrated by numerical simulations.

The remainder of the chapter is organized as follows. In Section II, notations and preliminary graph theory concepts used throughout the chapter are presented. Then the optimization problems and the distributed algorithm proposed to numerically solve them are described. In Section III, convergence analysis of the proposed algorithm is presented. The simulation results are subsequently provided in Section IV. Finally, Section V contains concluding remarks and directions for future work.

4.2 Problem Formulation

4.2.1 Preliminaries and Notation

Throughout this chapter, the set of real numbers greater than r is denoted by $\mathbb{R}_{>r}$, and the finite set of natural numbers $\{1, 2, ..., n\}$ is denoted by $\mathbb{N}_n$. The superscript T is used to indicate the transpose of a real vector or a matrix. The function $\mathrm{tr}(\cdot)$ denotes the trace of a given real matrix. Moreover, the inner product of two real matrices $\mathbf{v}$,$\mathbf{w} \in \mathbb{R}^{n\times n}$ is represented by $\langle \mathbf{v}, \mathbf{w} \rangle$ (note that $\langle \mathbf{v}, \mathbf{w} \rangle = \mathrm{tr}(\mathbf{v}\mathbf{w}^{\mathrm{T}}) = \sum_{i\in\mathbb{N}_n} \sum_{j\in\mathbb{N}_n} v_{ij} w_{ij}$). The ceiling function is represented by $\lceil . \rceil$, where for a real scalar $r \in \mathbb{R}$, $\lceil r \rceil$ gives the least integer greater than or equal to r. The real part of a complex number $c \in \mathbb{C}$ is denoted by $\Re(c)$.

Additionally, $\mathbb{B}_\sigma(\cdot)$ is a closed ball of radius $\sigma \in \mathbb{R}_{>0}$ centered around a given point, and $\mathbf{e}_{ij} \in \mathbb{R}^{n\times n}$ is a matrix whose elements are all zero, except for its (i, j) element which is equal to one. The domain of a given function is denoted by $\mathrm{dom}(\cdot)$, and the Frobenius norm of a given real matrix by $\|\cdot\|_{\mathrm{F}}$.

At any time instant $k \in \mathbb{N}$, let $G(k) = \big(V, E, \mathbf{W}(k)\big)$ denote a weighted directed graph (digraph) in the time interval $[t_k, t_{k+1})$, characterized by a set of vertices $V = \mathbb{N}_n$, a set of edges E, and a weight matrix $\mathbf{W}(k) \in \mathbb{R}^{n\times n}$. The (i, j) element of the weight matrix $\mathbf{W}(k)$, denoted by $w_{ij}(k)$, is the weight associated with the edge $\vec{ji} \in E$ for any pair of distinct nodes $i, j \in \mathbb{N}_n$ and any $k \in \mathbb{N}$. Note that $\vec{ji} \in E$ if node j sends information to node i at some point in time, i.e., $\vec{ji} \in E$ if $w_{ij}(k) \neq 0$ for some $k \in \mathbb{N}$. In the same time interval, the in-neighbor and out-neighbor sets associated with node i are defined as [2]

$$N_i^{in} = \{j \in V \setminus \{i\} | \vec{ji} \in E\}, \tag{4.1a}$$

$$N_i^{out} = \{j \in V \setminus \{i\} | \vec{ij} \in E\}, \tag{4.1b}$$

respectively. The Laplacian of the weighted digraph $G(k)$ is a real asymmetric matrix

$\mathbf{L}(k) \in \mathbb{R}^{n\times n}$ whose (i,j) element is given by [2, 45]

$$l_{ij}(k) = \begin{cases} -w_{ij}(k), & \text{if } \vec{ji} \in E, \\ \sum_{p\neq i} w_{ip}(k), & \text{if } j = i, \\ 0, & \text{otherwise}, \end{cases} \tag{4.2}$$

for any pair of distinct nodes $i, j \in \mathbb{N}_n$ and any $k \in \mathbb{N}$. The i^{th} eigenvalue of the Laplacian matrix $\mathbf{L}(k)$ is denoted by $\lambda_i(\mathbf{L}(k))$. The spectrum of a matrix is the set of all of its eigenvalues, and is denoted by $\Lambda(\cdot)$.

The generalized algebraic connectivity (GAC) of a weighted digraph $G(k)$ with Laplacian matrix $\mathbf{L}(k)$ is defined as the smallest real part of the nonzero eigenvalues of $\mathbf{L}(k)$, i.e.,

$$\tilde{\lambda}^{\mathbf{L}(k)} = \min_{\lambda_i(\mathbf{L}(k))\neq 0,\ \lambda_i(\mathbf{L}(k))\in\Lambda(\mathbf{L}(k))} \Re(\lambda_i(\mathbf{L}(k))), \tag{4.3}$$

for any $k \in \mathbb{N}$ [4]. In [4], it is shown that the imaginary part of the eigenvalue corresponding to the GAC is not related to the convergence rate of the distributed algorithms running on the network. Additionally, given expression (4.3), the GAC is defined only for a network where the second smallest eigenvalue of its graph Laplacian is nonzero. If a network has multiple connected components, but is not strongly connected as a whole, the GAC would only be applicable to each component separately.

4.2.2 Optimization Problems

Consider a time-varying asymmetric network with n stationary nodes, whose information exchange topology is represented by the weighted digraph $G(k)$ for all $k \in \mathbb{N}$. Similar to [29,52–54], it is assumed (as a more general formulation) that each node uses different power levels to communicate with its neighbors. However, it is not necessary to impose this assumption on every network, as for example in a large network, having different power levels for each out-neighbor is not feasible. The algorithm proposed in this study and the supporting analysis are not dependent on the assumption of a node using one or multiple transmission power levels communicate with its neighbors. The transmission power matrix is denoted by $\mathbf{P}(k) = [P_{ij}(k)] \in \mathbb{R}^{n\times n}$, for $i, j \in \mathbb{N}_n$ and $k \in \mathbb{N}$, where $P_{ij}(k) \in [P_{ij}^{\mathrm{low}}, P_{ij}^{\mathrm{up}}]$. In the literature, $P_{ij}(k)$ is used to denote the transmission power required by node i to transmit information to node j. In the present study, however, this order of indices is flipped in order to be consistent with the rest of the parameters used throughout the thesis, i.e., $P_{ij}(k)$ is the transmission power that node j uses to transmit information to node i. P_{ij}^{low} and P_{ij}^{up} are, respectively, the fixed lower and upper bounds of the permissible transmission power $P_{ij}(k)$, which are known *a priori*. It is assumed that the values of the transmission powers used by nodes for communication directly impact the existence probabilities of the network's communication links [1]. These probabilities can be regarded as the weight matrix $\mathbf{W}(k)$ of the network. Assuming a noise-limited environment, at any time instant $k \in \mathbb{N}$, the relation between the transmission power

$P_{ij}(k)$ and the link weight $w_{ij}(k)$ can be described by the following function

$$w_{ij}(k) = h(P_{ij}(k); \xi_{ij}), \tag{4.4}$$

for any $i \in \mathbb{N}_n$ and $j \in N_i^{in}$, where ξ_{ij} represents a set of real constant parameters characterizing the communication channel $\vec{ji}$, and $h(.;.)$ is a continuously differentiable and increasing function [29]. The stochastic framework for the existence probabilities mentioned above is encoded in $h(.;.)$ and can be found in [1]. To find the weight of a specific communication link experimentally, a number of messages are sent from the destination node to the target node, and the percentage of successfully received messages at the target node (when the received signal-to-noise ratio (SNR) is above a certain threshold) determines probability of the existence of that link.

Given the above formulation, the Laplacian matrix of the network can be expressed as a function of its transmission power. Consequently, the network GAC can be expressed as an implicit function of the power matrix, $\tilde{\lambda}(\mathbf{P}(k))$, for any $k \in \mathbb{N}$. Unlike the notion of algebraic connectivity introduced in [27] for undirected networks, an increase (or a decrease) in the elements of the transmission power matrix, which results in an increase (or a decrease) in the corresponding elements of the weight matrix $\mathbf{W}(k)$, does not necessarily lead to an increase (or a decrease) in the value of the network GAC [2]. This outcome is observed in the simulation results of this study as well. Additionally, unlike the algebraic connectivity of undirected networks which is concave [31], the GAC is neither concave nor convex, and hence, is referred to as a nonconvex function. Furthermore, the GAC is piecewise differentiable, but can be non-differentiable in certain points. As the elements of the transmission power matrix change, the discontinuity in the derivative of $\tilde{\lambda}(\mathbf{P}(k))$ occurs when the eigenvalue of the Laplacian matrix corresponding to the GAC changes from a real eigenvalue to a complex conjugate pair or vice versa. Ultimately, $\tilde{\lambda}(\mathbf{P}(k))$ is a locally Lipschitz continuous function with finite Lipschitz constants [2].

Given the transmission power vector $\mathrm{P}_i(k) = [P_{1i}(k), \cdots, P_{ni}(k)]^{\mathrm{T}}$ for any $i \in \mathbb{N}_n$ and $k \in \mathbb{N}$, the lifetime of node i is given by [21, 29, 32]

$$T(\mathrm{P}_i(k)) = \frac{\mathfrak{e}_i^0}{\sum_{j \in N_i^{out}} q_{ji} \cdot P_{ji}(k) \cdot \tau_{ji} \cdot K_{ji} + \sum_{j \in N_i^{in}} q_{ij} \cdot \mathfrak{e}_{ij}^r \cdot K_{ij}}, \tag{4.5}$$

where q_{ji} denotes the transmission rate per unit of time from node i to node j (like bits per second), τ_{ji} is the time that node i has to keep transmitting one packet of information with transmission power $P_{ji}(k)$ to ensure it has been received at node j without errors, and K_{ji} is the number of information packets sent from node i to node j. Moreover, $\mathfrak{e}_{ij}^r$ is the energy used by node i to receive information from node j, and $\mathfrak{e}_i^0$ is the initial energy of node i battery. This definition of lifetime indicates that if node i was to solely use a fixed transmission power vector $\mathrm{P}_i(k)$ for communication, its lifetime would be equal to $T(\mathrm{P}_i(k))$ given by (4.5). Considering the definitions of q_{ji} and τ_{ji}, the units of these parameters should be in line with each other from a practical point of view. Note that $T(\mathrm{P}_i(k))$ is a concave function on its domain and has a unit of time.

All parameters in (4.5), other than the transmission powers, are assumed to be constant and known. Furthermore, the energy required to sense incoming data is supposed to be negligible compared to transmission power requirements [17, 29]. Furthermore, the significant portion of the battery energy of a node is used to communicate with its neighbors [17, 21], and therefore, the energy that the nodes need to carry out the calculations required for the proposed algorithm is assumed to be negligble. Note that nodes only perform simple mathematical operations and need to use a basic consensus algorithm to determine any required global quantity. The lifetime definition given by (4.5) is relatively simple, indicating that a node's battery depletion has a linear relation with its transmission powers. It is known, however, that batteries have nonlinear dynamics in reality [17]. Nevertheless, the underlying optimization problem and the proposed algorithm are not critically dependent on the exact energy consumption model. The system (or network) lifetime is accordingly defined as the minimum lifetime over all nodes [17, 21], i.e., given the nodes have the transmission power vectors $\mathrm{P}_i(k)$, for all $i \in \mathbb{N}_n$ and any $k \in \mathbb{N}$, the network lifetime is

$$T_{sys}(\mathbf{P}(k)) = \min_{i \in \mathbb{N}_n} T(\mathrm{P}_i(k)). \tag{4.6}$$

In a noise-limited environment, a higher transmission power leads to stronger communication links and a more connected network [29]. However, that would be at the cost of higher total transmission power and lower node lifetime as evident from (4.5). Since a proper balance between the network lifetime, its connectivity level and the total transmission power is crucial for the efficient operation of the network, the following three optimization problems are considered for an asymmetric network:

$$\text{P1} \qquad \begin{aligned} \underset{\mathbf{P}}{\text{minimize}} \quad & -\tilde{\lambda}(\mathbf{P}) \\ \text{subject to} \quad & \sum_i \sum_j P_{ij}(k) \leq \bar{\mathrm{P}}, \\ & P_{ij}^{\text{low}} \leq P_{ij}(k) \leq P_{ij}^{\text{up}}, \end{aligned}$$

$$\text{P2} \qquad \begin{aligned} \underset{\mathbf{P}}{\text{minimize}} \quad & \sum_i \sum_j P_{ij} \\ \text{subject to} \quad & \tilde{\lambda}(\mathbf{P}(k)) \geq \underline{\lambda}, \\ & P_{ij}^{\text{low}} \leq P_{ij}(k) \leq P_{ij}^{\text{up}}, \end{aligned}$$

$$\text{P3} \qquad \begin{aligned} \underset{\mathbf{P}}{\text{minimize}} \quad & -T_{sys}(\mathbf{P}) \\ \text{subject to} \quad & \tilde{\lambda}(\mathbf{P}(k)) \geq \underline{\lambda}, \\ & P_{ij}^{\text{low}} \leq P_{ij}(k) \leq P_{ij}^{\text{up}}, \end{aligned}$$

for all $i \in \mathbb{N}_n, j \in N_i^{in}$ and $k \in \mathbb{N}$, where $\bar{\mathrm{P}}$ and $\underline{\lambda}$ are prespecified bounds, reflecting the highest acceptable total transmission power and the lowest connectivity level of the network, respectively. Given the number of constraints, which at most is equal to $m = 2(n^2 - n) + 1$, and that $\tilde{\lambda}(\mathbf{P}(k))$ is an implicit nonlinear function of the transmission power matrix of the network, finding analytical solutions for the optimization problems P1-P3 may not be feasible.

In the optimization problem P1, it is desired to maximize the connectivity level of the network while considering an upper limit for the total transmission power consumed by the network (the first constraint) and limiting the transmission power of each node to a prespecified range (the second set of constraints). In the optimization problem P2, the objective is to minimize the total transmission power of the network while considering a lower limit for the connectivity level of the network (the first constraint) and again limiting the transmission powers to a prescribed range. In optimization problem P3, it is desired to maximize the network lifetime with the same set of constraints as the optimization problem P2. The three problems are summarized in Table 4.1. Each of the three optimization problems introduced above can play an important role in underwater acoustic sensor networks in different periods of time. For example, consider a sensor network deployed in a noisy environment prior to winter, meant to last long. Initially, in adverse noise conditions, the GAC needs to be maximized while there is no immediate concern of the new battery being depleted; this obviously relates to the framework of the first optimization problem. As winter settles, the water temperature drop will have a negative impact on the battery reserve capacity, which means that now the total network transmission power must be managed while maintaining minimal network connectivity, which matches the second optimization problem. After many

Table 4.1: The proposed optimization problems.

Optimization Problem	GAC	Transmission Power	Network Lifetime
P1	Maximize	Constraint (Individual Nodes and Total of Network)	-
P2	Constraint	Minimize (Total of Network) Constraint (Nodes)	-
P3	Constraint	Constraint (Nodes)	Maximize

months of operation, battery reserves are such that now the overall network lifetime is at risk and must be maximized; this can be described by the third optimization problem. The optimization problem P3 with additional constraints $T(\mathrm{P}_i(k)) = T_{sys}(\mathbf{P}(k))$, for all $i \in \mathbb{N}_n$, is investigated in [29] using a centralized approach. These lifetime constraints are not considered in this study in order to achieve a more streamlined approach using only the interior point method for all three optimization problems. A centralized solution to the optimization problem P2 was proposed in [28]. However, unlike the current work, in [28], each node uses just a single transmission power level to communicate with its neighbors, i.e., the concatenation of all transmission powers is a vector, not a matrix as in this study. In the sequel, distributed algorithms are proposed to numerically find a local minima of the above-mentioned problems.

Let $l \in \mathbb{N}_3$ be an index used to distinguish between the three optimization problems, with $l = 1, 2, 3$ representing P1, P2 and P3, respectively. For all $i \in \mathbb{N}_n$, $j \in N_i^{in}$

and $k \in \mathbb{N}$, define

$$f_l(\mathbf{P}(k)) = \begin{cases} -\tilde{\lambda}(\mathbf{P}(k)), & \text{if} \quad l = 1, \\ \sum_i \sum_j P_{ij}(k), & \text{if} \quad l = 2, \\ -T_{sys}(\mathbf{P}(k)), & \text{if} \quad l = 3, \end{cases}$$

as the main cost functions, and

$$h_l^1(k) = \begin{cases} \bar{\mathrm{P}} - \sum_i \sum_j P_{ij}(k), & \text{if} \quad l = 1, \\ \tilde{\lambda}(\mathbf{P}(k)) - \underline{\lambda}, & \text{if} \quad l = 2 \text{ or } 3, \end{cases}$$

$$h_{ij}^2(k) = P_{ij}^{\text{up}} - P_{ij}(k),$$

$$h_{ij}^3(k) = P_{ij}(k) - P_{ij}^{\text{low}},$$

as the constraint functions. Since the equality constraints in [29] are not considered in the problem P3, the interior point method, similar to [28], is utilized to transform the inequality-constrained optimization problems P1-P3 into sequential unconstrained problems. To this end, depending on the value of l, minimizers of the following problems converge asymptotically to the respective minimizers of the optimization problems P1-P3,

$$\underset{\mathbf{P}}{\text{minimize}} \quad \tilde{f}_l(\mathbf{P}, \Gamma), \tag{4.7}$$

where

$$\tilde{f}_l(\mathbf{P}(k), \Gamma) = f_l(\mathbf{P}(k)) + \Gamma^{-1} \cdot I_l(\mathbf{P}(k)), \tag{4.8}$$

$$I_l(\mathbf{P}(k)) = -\log\left(h_l^1(k)\right) - \sum_{i\in\mathbb{N}_n}\sum_{j\in\mathbb{N}_i^{in}} \log\left(h_{ij}^2(k)\right) - \sum_{i\in\mathbb{N}_n}\sum_{j\in\mathbb{N}_i^{in}} \log\left(h_{ij}^3(k)\right), \tag{4.9}$$

and $\Gamma \in \mathbb{R}_{>0}$ is updated as the optimization algorithm proceeds [13, 14]. For any $l \in \mathbb{N}_3$, the set $\mathcal{R}_l$ is defined as the domain of the corresponding joint cost function in (4.8). The general procedure to numerically solve the sequential unconstrained optimization problem (4.7) is presented in Algorithm 4.1, where the initial transmission power matrix $\mathbf{P}^0 = \mathbf{P}(0)$ needs to be strictly inside the corresponding feasible set $\mathcal{R}_l$ and not on its boundaries [28, 29]. There are three design parameters in Algorithm 4.1, namely ϵ, μ and Γ^0, which need to be chosen appropriately. The choice of ϵ involves a trade-off between the accuracy and the execution speed of the algorithm, whereas the parameter μ determines the rate of increase of Γ and will be discussed later. The parameter Γ^0 denotes the initial weight given to the penalty function $I_l(\mathbf{P}(k))$.

To find $\mathbf{P}^*$ in line 4 of Algorithm 4.1, the subgradient method with backtracking line search is utilized. The subgradient method enables the procedure to deal with the non-differentiable cost function $\tilde{f}_l(\mathbf{P}(k), \Gamma)$, whereas the backtracking line search gives the maximum allowable step size to move along the search direction obtained via the subgradient approach. In [28, 29], the same approach was used to solve the considered optimization problems using a centralized approach. The main focus of this chapter, however, is to propose a *distributed* algorithm, while taking into consideration the estimation of global variables. Note that since the GAC is a nonconvex function, the joint cost function will also be nonconvex. The definitions of a matrix-valued function's

Algorithm 4.1. A procedure for solving the optimization problem (4.7) for $l = 1, 2, 3$.

1: Given strictly feasible $\mathbf{P} = \mathbf{P}^0 \in \mathbb{R}^{n\times n}$, initialize $\Gamma = \Gamma^0 \in \mathbb{R}_{>0}$.
2: Choose arbitrary constants $\mu \in \mathbb{R}_{>1}$ and $\epsilon \in \mathbb{R}_{>0}$.
3: **while** $m\Gamma^{-1} > \epsilon$
4: Compute $\mathbf{P}^* \in \underset{\mathbf{P}\in\mathcal{R}_l}{\operatorname{argmin}}\ \tilde{f}_l(\mathbf{P}, \Gamma)$.
5: $\mathbf{P} = \mathbf{P}^*$
6: $\Gamma = \mu\Gamma$
7: **end while**

subgradient and supergradient are given next (the same definition was used in [3] but is brought here for the sake of self-containedness).

Definition 4.1. Matrix $\mathbf{g}$ is said to be the subgradient of the nonconvex and non-differentiable function $\tilde{f}_l : \mathbb{R}^{n\times n} \to \mathbb{R}$ at $\mathbf{P} \in \mathcal{R}_l$ if there exists a real scalar $\sigma \in \mathbb{R}_{>0}$ such that for any $\hat{\mathbf{P}} \in \mathbb{B}_\sigma(\mathbf{P})$, the following inequality holds:

$$\tilde{f}_l(\hat{\mathbf{P}}, \Gamma) \geq \tilde{f}_l(\mathbf{P}, \Gamma) + \langle \hat{\mathbf{P}} - \mathbf{P}, \mathbf{g} \rangle. \tag{4.10}$$

The set of all subgradients of function $\tilde{f}_l$ at $\mathbf{P} \in \mathcal{R}_l$ is called the subdifferential set $\partial\tilde{f}_l(\mathbf{P}, \Gamma)$ [2, 14].

Definition 4.2. For the nonconvex and non-differentiable function $\tilde{\lambda} : \mathbb{R}^{n\times n} \to \mathbb{R}$, matrix $\mathbf{g}'$ is said to be its supergradient at $\mathbf{P} \in \operatorname{dom}(\tilde{\lambda})$ if there exists a real scalar $\sigma \in \mathbb{R}_{>0}$ such that for any $\hat{\mathbf{P}} \in \mathbb{B}_\sigma(\mathbf{P})$, the following inequality holds:

$$\tilde{\lambda}(\hat{\mathbf{P}}) \leq \tilde{\lambda}(\mathbf{P}) + \langle \hat{\mathbf{P}} - \mathbf{P}, \mathbf{g}' \rangle. \tag{4.11}$$

The set of all supergradients of function $\tilde{\lambda}$ at $\mathbf{P} \in \mathrm{dom}(\tilde{\lambda})$ is called the superdifferential set $\partial\tilde{\lambda}(\mathbf{P})$ [2].

Using the subgradient method, the current iteration of the optimization loop is moved in the opposite direction of a subgradient of the cost function ($\tilde{f}_l(\mathbf{P}(k), \Gamma)$ for a specific l). That is, at any time t_k, $k \in \mathbb{N}$,

$$\mathbf{P}(k+1) = \mathbf{P}(k) + \alpha(k) \cdot \mathbf{v}(k), \tag{4.12}$$

where $\mathbf{v}(k) = -\mathbf{g}(k)$ is the search direction, and $\mathbf{g}(k) = [g_{ij}(k)] \in \partial\tilde{f}_l(\mathbf{P}(k), \Gamma)$, for $i \in \mathbb{N}_n$, $j \in N_i^{in}$ and a specific $l \in \mathbb{N}_3$, is an arbitrary subgradient of the cost function $\tilde{f}_l(\mathbf{P}(k), \Gamma)$. Also, $\alpha(k)$ is the step size obtained via the backtracking line search, determining the magnitude of the move along the search direction. For different l, at any time t_k with $k \in \mathbb{N}$, and for any $i \in \mathbb{N}_n$ and $j \in N_i^{in}$, the $(i,j)^{\text{th}}$ element of $\mathbf{g}(k)$ is given by

- if $l = 1$,

$$g_{ij}(k) = -\nabla\tilde{\lambda}_{ij}(\mathbf{P}(k)) + \Gamma^{-1}\Big(\frac{1}{\bar{\mathrm{P}} - \sum_i \sum_j P_{ij}(k)} + \frac{1}{P_{ij}^{\text{up}} - P_{ij}(k)} - \frac{1}{P_{ij}(k) - P_{ij}^{\text{low}}}\Big), \tag{4.13}$$

- if $l = 2$,

$$g_{ij}(k) = 1 + \Gamma^{-1}\Big(- \frac{\nabla\tilde{\lambda}_{ij}(\mathbf{P}(k))}{\tilde{\lambda}(\mathbf{P}(k)) - \underline{\lambda}} + \frac{1}{P_{ij}^{\text{up}} - P_{ij}(k)} - \frac{1}{P_{ij}(k) - P_{ij}^{\text{low}}}\Big), \tag{4.14}$$

- if $l = 3$,

$$g_{ij}(k) = -\frac{\partial T_{sys}(\mathbf{P}(k))}{\partial P_{ij}(k)} + \Gamma^{-1}\Big(- \frac{\nabla\tilde{\lambda}_{ij}(\mathbf{P}(k))}{\tilde{\lambda}(\mathbf{P}(k)) - \underline{\lambda}} + \frac{1}{P_{ij}^{\text{up}} - P_{ij}(k)} - \frac{1}{P_{ij}(k) - P_{ij}^{\text{low}}}\Big), \tag{4.15}$$

where $\nabla\tilde{\lambda}_{ij}(\mathbf{P}(k))$ is the (i,j) element of the subgradient matrix of the GAC, i.e., the first-order partial derivative of the GAC with respect to $P_{ij}(k)$, and is arbitrarily chosen from the subdifferential set [28, 29]

$$\partial\tilde{\lambda}_{ij}(\mathbf{P}(k)) \coloneqq \frac{1}{\beta(k)}\{\partial\tilde{\lambda}_{ij}^{+}(\mathbf{P}(k)), \partial\tilde{\lambda}_{ij}^{-}(\mathbf{P}(k))\}. \tag{4.16}$$

In (4.16), $\beta(k) = \beta_0/k^{\beta_0}$, for some $\beta_0 \in \mathbb{R}_{>0}$, is the step size used to numerically compute $\partial\tilde{\lambda}_{ij}^{+}(\mathbf{P}(k))$ and $\partial\tilde{\lambda}_{ij}^{-}(\mathbf{P}(k))$ given by [2, 28, 29]

$$\partial\tilde{\lambda}_{ij}^{+}(\mathbf{P}(k)) = \tilde{\lambda}(\mathbf{P}(k) + \beta(k)\mathbf{e}_{ij}) - \tilde{\lambda}(\mathbf{P}(k)), \tag{4.17a}$$

$$\partial\tilde{\lambda}_{ij}^{-}(\mathbf{P}(k)) = \tilde{\lambda}(\mathbf{P}(k)) - \tilde{\lambda}(\mathbf{P}(k) - \beta(k)\mathbf{e}_{ij}). \tag{4.17b}$$

In (4.13)-(4.15), let $g_{ij}^{+}(k)$ and $g_{ij}^{-}(k)$, for any $i \in \mathbb{N}_n$, $j \in N_i^{in}$ and $k \in \mathbb{N}$, correspond to the cases where $\nabla\tilde{\lambda}_{ij}(\mathbf{P}(k))$ has been explicitly chosen to be equal to $\frac{1}{\beta(k)}\partial\tilde{\lambda}_{ij}^{+}(\mathbf{P}(k))$ or $\frac{1}{\beta(k)}\partial\tilde{\lambda}_{ij}^{-}(\mathbf{P}(k))$, respectively. If no superscript is used in the notation of $g_{ij}(k)$, it means that the choice of $\nabla\tilde{\lambda}_{ij}(\mathbf{P}(k))$ is arbitrary. To compute the partial derivatives of the network lifetime for each $k \in \mathbb{N}$ in (4.15), one needs to first determine which node the minimum lifetime corresponds to before using (4.6).

The general procedure to implement (4.12), i.e. line 4 of Algorithm 4.1, is presented in Fig. 4.1. The initialization of the variables is not shown in the figure as it happens in the previous lines of Algorithm 4.1. This inner loop is repeated for a prespecified number of iterations before moving to line 5 of Algorithm 4.1. If the outcome of a decision block is No, the inner loop is terminated, and the parameter μ is updated as described later. Then the optimization algorithm skips line 5 and moves to line 6. The steps in Fig. 4.1 need to be implemented in a distributed manner, and will be discussed in further detail in the next section. For this purpose, the following assumptions are required.

Assumption 4.1. *The network digraph is assumed to be strongly connected at all times, meaning that there is a directed path from every node in the graph to every other node.*

Assumption 4.2. *As the elements of the transmission power matrix $\boldsymbol{P}(k)$ vary within the permissible set $\mathcal{R}_l$, for $l = 1, 2, 3$, the weighted digraph $G(k)$ remains structurally static, i.e., no edges are added or removed during the optimization process.*

Remark 4.1. Under the following two conditions, it is guaranteed that the digraph is structurally static, as required in Assumption 4.2.

- $P_{ij}^{\text{low}} \in \mathbb{R}_{>0}$, for all $i \in \mathbb{N}_n$ and $j \in N_i^{in}$, should be chosen such that if a transmission power of a node is equal to this value, the corresponding weight, i.e. the existence probability of that communication link, is strictly greater than zero. This ensures that no communication link is removed.

- If the transmission power corresponding to a communication link is initially zero, i.e., if a communication link does not initially exist, it will not be considered in the optimization algorithm. Therefore, any possible addition of new edges is disregarded.

Assumption 4.3. *To be able to estimate the GAC (which is a global quantity) using only local information, the eigenvalue representing the GAC should be observable to the nodes [33]. It is assumed that the initial topology of the network satisfies the necessary conditions for observability, as mentioned in [34, 35].*

Remark 4.2. Given that the topology of the graph is supposed to be invariant (Assumption 4.2), and that GAC is assumed to be observable in the initial topology, the GAC remains observable during the entire optimization procedure.

4.3 Distributed optimization algorithm

In this section, distributed implementation of the elements of the inner optimization loop (demonstrated in Fig. 4.1), is investigated, and the proposed distributed optimization algorithm is presented at the end.

Starting with the distributed calculation of the search direction $\mathbf{v}(k)$, it can be easily understood from (4.13)-(4.17) that each node requires knowledge of certain global variables in order to determine the direction along which to update its transmission

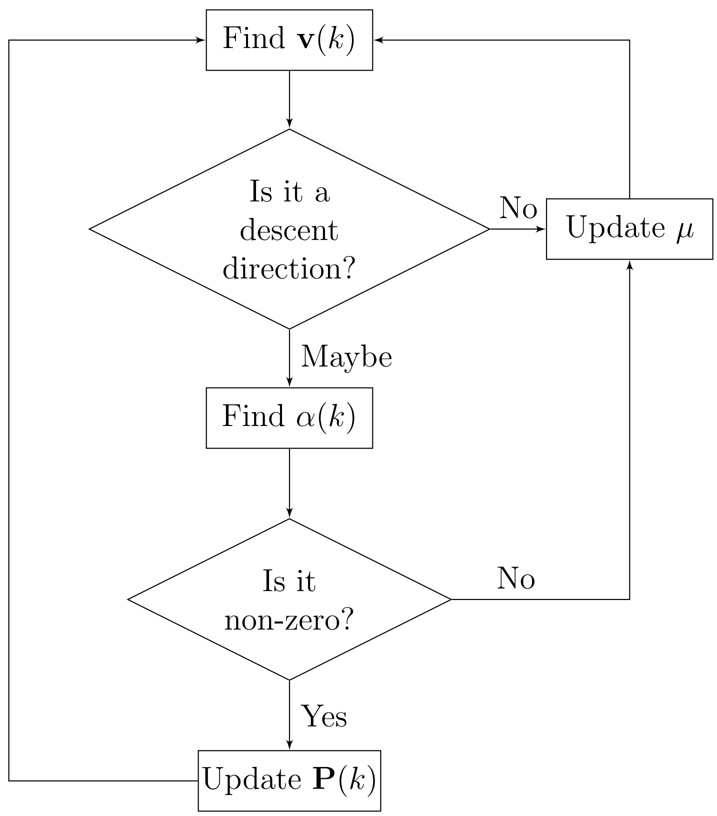

Figure 4.1: The inner optimization loop (line 4 of Algorithm 4.1).

powers. That is, for all $i \in \mathbb{N}_n$, $j \in N_i^{in}$ and any $k \in \mathbb{N}$, the following global variables are present in (4.13)-(4.17): $\tilde{\lambda}(\mathbf{P}(k))$, $\tilde{\lambda}(\mathbf{P}(k) + \beta(k)\mathbf{e}_{ij})$ or $\tilde{\lambda}(\mathbf{P}(k) - \beta(k)\mathbf{e}_{ij})$ for the calculation of $g_{ij}(k)$. It is relatively easy to obtain the sum of transmission powers $\sum_i \sum_j P_{ij}(k)$ and the partial derivatives of the network lifetime $\partial T_{sys}(\mathbf{P}(k))/\partial P_{ij}(k)$ as they only require a simple consensus between the nodes. On the contrary, calculating the GAC using only local information for each $k \in \mathbb{N}$ is challenging. For this purpose, the methods of [4] and [33] can be utilized within the proposed optimization algorithm. The algorithm of [4] converges to an arbitrarily close neighborhood of the value of the GAC, whereas the algorithm of [33] can be used to obtain the exact values of the observable eigenvalues of the graph Laplacian, including the eigenvalue corresponding to the GAC, in finite-time. Due to Assumption 4.3, the results of [4] and [33] can be used to obtain the GAC, but the approach of [33] will be utilized in this work.

From the previous paragraph, one may insinuate that the distributed calculation of the elements of $\mathbf{v}(k)$ is straightforward; however, unlike the gradient method, the search direction obtained via the subgradient method does not necessarily yield a descent direction at every iteration $k \in \mathbb{N}$. Consequently, the first decision-making block in Fig. 4.1 helps to determine if $\mathbf{v}(k)$ fails to be a descent direction at time t_k based on the following result.

Lemma 4.1. *For the non-differentiable function $\tilde{f}_l(\boldsymbol{P}(k), \Gamma)$ with subdifferential set*

$\partial \tilde{f}_l(\boldsymbol{P}(k), \Gamma)$ *at* $\boldsymbol{P}(k) \in \mathcal{R}_l$, $l = 1, 2, 3$, $\boldsymbol{v}(k)$ *is not a descent direction if*

$$\exists\ \boldsymbol{g}(k) \in \partial \tilde{f}_l(\boldsymbol{P}(k), \Gamma) \text{ such that } \langle \boldsymbol{v}(k), \boldsymbol{g}(k) \rangle \geq 0, \tag{4.18}$$

for any $k \in \mathbb{N}$ *and* $\Gamma \in \mathbb{R}_{>0}$.

Proof. The proof follows directly from the definition of the subgradient (Definition 4.1). ■

Note that Lemma 4.1 is presented in [29] and is brought here for the sake of completeness. Also, as mentioned in [29], Lemma 4.1 does not provide a necessary condition for $\mathbf{v}(k)$ to be a descent direction, but rather a sufficient one for not to be a descent direction (this is explained in detail later). To implement Lemma 4.1 in a distributed manner, $\langle \mathbf{v}(k), \mathbf{g}(k) \rangle$, for each $l \in \mathbb{N}_3$, expands in a distributed manner as follows

- if $l = 1$,

$$\begin{aligned}\langle \mathbf{v}(k), \mathbf{g}(k) \rangle = \sum_{i \in \mathbb{N}_n} \sum_{j \in N_i^{in}} \Big(&- \nabla \tilde{\lambda}_{ij}^1(\mathbf{P}(k)) \cdot \nabla \tilde{\lambda}_{ij}^2(\mathbf{P}(k)) \\ &+ \Gamma^{-1} \big(\nabla \tilde{\lambda}_{ij}^1(\mathbf{P}(k)) + \nabla \tilde{\lambda}_{ij}^2(\mathbf{P}(k)) \big) \frac{\partial I_1(\mathbf{P}(k))}{\partial P_{ij}(k)} - \Gamma^{-2} \big(\frac{\partial I_1(\mathbf{P}(k))}{\partial P_{ij}(k)} \big)^2 \Big),\end{aligned} \tag{4.19}$$

- if $l = 2$,

$$\langle \mathbf{v}(k), \mathbf{g}(k) \rangle = \sum_{i \in \mathbb{N}_n} \sum_{j \in N_i^{in}} \Big(-1 - \Gamma^{-1} \big(\frac{\partial I_2^1(\mathbf{P}(k))}{\partial P_{ij}(k)} + \frac{\partial I_2^2(\mathbf{P}(k))}{\partial P_{ij}(k)} \big) - \Gamma^{-2} \frac{\partial I_2^1(\mathbf{P}(k))}{\partial P_{ij}(k)} \cdot \frac{\partial I_2^2(\mathbf{P}(k))}{\partial P_{ij}(k)} \Big), \quad (4.20)$$

- if $l = 3$,

$$\langle \mathbf{v}(k), \mathbf{g}(k) \rangle = \sum_{i \in \mathbb{N}_n} \sum_{j \in N_i^{in}} \Big(- \big(\frac{\partial T_{sys}(\mathbf{P}(k))}{\partial P_{ij}(k)} \big)^2 + \Gamma^{-1} \frac{\partial T_{sys}(\mathbf{P}(k))}{\partial P_{ij}(k)} \big(\frac{\partial I_3^1(\mathbf{P}(k))}{\partial P_{ij}(k)} + \frac{\partial I_3^2(\mathbf{P}(k))}{\partial P_{ij}(k)} \big) - \Gamma^{-2} \frac{\partial I_3^1(\mathbf{P}(k))}{\partial P_{ij}(k)} \cdot \frac{\partial I_3^2(\mathbf{P}(k))}{\partial P_{ij}(k)} \Big), \quad (4.21)$$

where $\partial I_l(\mathbf{P}(k))/\partial P_{ij}(k)$ is the partial derivative of the indicator function (4.9), and corresponds to the terms inside the parenthesis multiplied by Γ^{-1} in (4.13)-(4.15). The numerical superscripts in the variables of (4.19)-(4.21) correspond to the (possibly) different supergradients of the GAC used in obtaining those expressions. That is, $\mathbf{v}(k)$ and $\mathbf{g}(k)$ are not necessarily the same for distinct values of l. In the distributed implementation of (4.19)-(4.21), the worst-case scenario is considered for $\mathbf{g}(k)$, i.e., for each $i \in \mathbb{N}_n$, $j \in N_i^{in}$, $\sum_i \sum_j P_{ij}(k)$, $\partial T_{sys}(\mathbf{P}(k))/\partial P_{ij}(k)$, $\tilde{\lambda}(\mathbf{P}(k))$ and the values of both $\tilde{\lambda}(\mathbf{P}(k) + \beta(k)\mathbf{e}_{ij})$ and $\tilde{\lambda}(\mathbf{P}(k) - \beta(k)\mathbf{e}_{ij})$ are estimated, and are checked to see which one results in the most positive $v_{ij}(k) \cdot g_{ij}(k)$ element. Note that Lemma 4.1 is implemented in a distributed manner for $l = 1, 2, 3$ as per Algorithm 4.2.

As mentioned before, finding $\sum_i \sum_j P_{ij}(k)$ or $\partial T_{sys}(\mathbf{P}(k))/\partial P_{ij}(k)$ in Algorithm 4.2 is simple. On the other hand, more complex algorithms are required to obtain the GAC in a distributed way. Using the finite-time method of [33], the obtained GAC values in different optimization iterations will be exact at every node, meaning that Algorithm 4.2 can be implemented as it is. However, if the method of [4] is used, the estimated GAC values at each node will be within an arbitrarily small neighborhood of the exact values. The resulting errors would need to be taken into consideration as they may lead to erroneous decisions/values throughout the algorithm. In the sequel, only the exact method of [33] is exploited.

As noted earlier, the search direction $\mathbf{v}(k)$ may not be a descent direction for some $k \in \mathbb{N}$. In this case, as per Algorithm 4.1 and Fig. 4.1, the optimization algorithm is terminated at iteration k without updating the transmission power matrix $\mathbf{P}(k)$ or increasing the optimization iteration index k. The value of Γ is then increased. The repetition of the k^{th} optimization iteration is denoted by k', where the only difference between this and the k^{th} iteration is the updated value for Γ ($\Gamma' = \mu\Gamma$). It is now desired to update μ such that the non-descent search direction of iteration k will be a descent one at iteration k' (hence the μ update block in Fig. 4.1).

Lemma 4.2. *For each optimization problem (4.7), if $\boldsymbol{v}(k)$ is not a descent direction for some $k \in \mathbb{N}$ according to Lemma 4.1, the value of μ used to update Γ should satisfy the following inequalities in order to have a descent direction at iteration k'*

Algorithm 4.2. Distributed implementation of Lemma 4.1.

1: Nodes estimate $\tilde{\lambda}(\mathbf{P}(k))$ using the information locally.
2: For every $i \in \mathbb{N}_n$ and $j \in N_i^{in}$, the values of $\tilde{\lambda}(\mathbf{P}(k) + \beta(k)\mathbf{e}_{ij})$ and $\tilde{\lambda}(\mathrm{P}(k) - \beta(k)\mathbf{e}_{ij})$ are estimated using information locally available to node i.
3: Depending on the optimization problem P1, P2 or P3, nodes communicate to obtain $\sum_i \sum_j P_{ij}(k)$ or $\partial T_{sys}(\mathbf{P}(k))/\partial P_{ij}(k)$.
4: Each node calculates both $g_{ij}^{+}(k)$ and $g_{ij}^{-}(k)$, where $v_{ij}(k)$ is equal to one of the two.
5: Each node calculates $x_{ij}^1 = v_{ij}(k) \cdot g_{ij}^{-}(k)$ and $x_{ij}^2 = v_{ij}(k) \cdot g_{ij}^{+}(k)$.
5: $x_{ij} = \max\{x_{ij}^1, x_{ij}^2\}$
6: Nodes communicate to calculate $\mathcal{X} = \sum_i \sum_j x_{ij}$.
7: If $\mathcal{X} \geq 0$, then $\mathbf{v}(k)$ is not a descent direction.

- *for* $l = 1$,

$$\mu \geq \lceil \frac{\sum_{i\in\mathbb{N}_n} \sum_{j\in N_i^{in}} \left(\nabla\tilde{\lambda}_{ij}^1(\boldsymbol{P}(k)) + \nabla\tilde{\lambda}_{ij}^2(\boldsymbol{P}(k))\right) \frac{\partial I_1(\boldsymbol{P}(k))}{\partial P_{ij}(k)}}{\Gamma \sum_{i\in\mathbb{N}_n} \sum_{j\in N_i^{in}} \nabla\tilde{\lambda}_{ij}^1(\boldsymbol{P}(k)) \cdot \nabla\tilde{\lambda}_{ij}^2(\boldsymbol{P}(k))} - 1\rceil, \tag{4.22}$$

- *for* $l = 2$,

$$\mu \geq \lceil \frac{-1}{n^2\Gamma} \sum_{i\in\mathbb{N}_n} \sum_{j\in N_i^{in}} \left(\frac{\partial I_2^1(\boldsymbol{P}(k))}{\partial P_{ij}(k)} + \frac{\partial I_2^2(\boldsymbol{P}(k))}{\partial P_{ij}(k)}\right) - 1\rceil, \tag{4.23}$$

- *for* $l = 3$,

$$\mu \geq \lceil \frac{\sum_{i\in\mathbb{N}_n} \sum_{j\in N_i^{in}} \frac{\partial T_{sys}(\boldsymbol{P}(k))}{\partial P_{ij}(k)} \left(\frac{\partial I_3^1(\boldsymbol{P}(k))}{\partial P_{ij}(k)} + \frac{\partial I_3^2(\boldsymbol{P}(k))}{\partial P_{ij}(k)}\right)}{\Gamma \sum_{i\in\mathbb{N}_n} \sum_{j\in N_i^{in}} \left(\frac{\partial T_{sys}(\boldsymbol{P}(k))}{\partial P_{ij}(k)}\right)^2} - 1\rceil, \tag{4.24}$$

where the numerical superscripts in the variables of (4.22)-(4.24) are defined as before.

Proof. For $l = 1$, according to Lemma 4.1, if $\exists\, \mathbf{g}(k) \in \partial\tilde{f}_1(\mathbf{P}(k), \Gamma)$ such that $\langle \mathbf{v}(k), \mathbf{g}(k)\rangle \geq$

0, then $\mathbf{v}(k)$ is not a descent direction at iteration k, for some $k \in \mathbb{N}$. By terminating the inner optimization loop and updating the value of Γ, at iteration k', one obtains

$$\langle \mathbf{v}(k'), \mathbf{g}(k') \rangle = \sum_{i\in\mathbb{N}_n} \sum_{j\in N_i^{in}} \Big(- \nabla\tilde{\lambda}_{ij}^1(\mathbf{P}(k)) \cdot \nabla\tilde{\lambda}_{ij}^2(\mathbf{P}(k)) + \\ \Gamma'^{-1}\big(\nabla\tilde{\lambda}_{ij}^1(\mathbf{P}(k)) + \nabla\tilde{\lambda}_{ij}^2(\mathbf{P}(k))\big)\frac{\partial I_1(\mathbf{P}(k))}{\partial P_{ij}(k)} - \Gamma'^{-2}\big(\frac{\partial I_1(\mathbf{P}(k))}{\partial P_{ij}(k)}\big)^2\Big), \quad (4.25)$$

where $\Gamma' = \mu\Gamma$ as previously mentioned. It is desired to make the inner product (4.25) less than zero by appropriately choosing the value of μ that Γ, at iteration k, is multiplied with. To this end, the right-hand side of (4.19) is multiplied by μ^{-2}, and is re-arranged to obtain

$$\sum_{i\in\mathbb{N}_n} \sum_{j\in N_i^{in}} \Big(- \nabla\tilde{\lambda}_{ij}^1(\mathbf{P}(k)) \cdot \nabla\tilde{\lambda}_{ij}^2(\mathbf{P}(k)) + \\ \mu^{-1}\Gamma^{-1}\big(\nabla\tilde{\lambda}_{ij}^1(\mathbf{P}(k)) + \nabla\tilde{\lambda}_{ij}^2(\mathbf{P}(k))\big)\frac{\partial I_1(\mathbf{P}(k))}{\partial P_{ij}(k)} - \mu^{-2}\Gamma^{-2}\big(\frac{\partial I_1(\mathbf{P}(k))}{\partial P_{ij}(k)}\big)^2\Big) \leq \\ \sum_{i\in\mathbb{N}_n} \sum_{j\in N_i^{in}} \Big(- \frac{\mu^2+1}{\mu^2}\nabla\tilde{\lambda}_{ij}^1(\mathbf{P}(k)) \cdot \nabla\tilde{\lambda}_{ij}^2(\mathbf{P}(k)) + \\ \frac{\mu+1}{\mu^2}\Gamma^{-1}\big(\nabla\tilde{\lambda}_{ij}^1(\mathbf{P}(k)) + \nabla\tilde{\lambda}_{ij}^2(\mathbf{P}(k))\big)\frac{\partial I_1(\mathbf{P}(k))}{\partial P_{ij}(k)} - \frac{2}{\mu^2}\Gamma^{-2}\big(\frac{\partial I_1(\mathbf{P}(k))}{\partial P_{ij}(k)}\big)^2\Big). \quad (4.26)$$

The left-hand side of (4.26) is equal to (4.25), and as mentioned, it is desired to be less than zero. To guarantee this, the right-hand side of (4.26) needs to be less than zero.

From (4.19), one has

$$\sum_{i\in\mathbb{N}_n}\sum_{j\in N_i^{in}}\Gamma^{-2}\Big(\frac{\partial I_1(\mathbf{P}(k))}{\partial P_{ij}(k)}\Big)^2 \le \sum_{i\in\mathbb{N}_n}\sum_{j\in N_i^{in}}\Big(-\nabla\tilde{\lambda}^1_{ij}(\mathbf{P}(k))\cdot\nabla\tilde{\lambda}^2_{ij}(\mathbf{P}(k))+$$
$$\Gamma^{-1}\big(\nabla\tilde{\lambda}^1_{ij}(\mathbf{P}(k))+\nabla\tilde{\lambda}^2_{ij}(\mathbf{P}(k))\big)\frac{\partial I_1(\mathbf{P}(k))}{\partial P_{ij}(k)}\Big). \quad (4.27)$$

By incorporating the inequality (4.27) in the right-hand side of (4.26) and simplifying, one arrives at

$$\mu \ge \Big\lceil \frac{\sum_{i\in\mathbb{N}_n}\sum_{j\in N_i^{in}}\big(\nabla\tilde{\lambda}^1_{ij}(\mathbf{P}(k))+\nabla\tilde{\lambda}^2_{ij}(\mathbf{P}(k))\big)\frac{\partial I_1(\mathbf{P}(k))}{\partial P_{ij}(k)}}{\Gamma\sum_{i\in\mathbb{N}_n}\sum_{j\in N_i^{in}}\nabla\tilde{\lambda}^1_{ij}(\mathbf{P}(k))\cdot\nabla\tilde{\lambda}^2_{ij}(\mathbf{P}(k))} - 1\Big\rceil,$$

which guarantees $\langle \mathbf{v}(k'), \mathbf{g}(k')\rangle < 0$. This completes the proof for $l = 1$. For $l = 2, 3$, the proofs follow a similar argument. ■

The implementation of Algorithm 4.2 precedes Lemma 4.2. Therefore, if the nodes need to update the value of μ as per Lemma 4.2, they have already obtained the required global values in Algorithm 4.2, and only need to use a basic consensus algorithm to determine the summations in (4.22)-(4.24).

Recall that Lemma 4.1 provides a sufficient condition for not having a descent search direction. That is, at any time t_k, $k \in \mathbb{N}$, even if $\langle \mathbf{v}(k), \mathbf{g}(k)\rangle < 0$ for all $\mathbf{g}(k) \in \partial\tilde{f}_l(\mathbf{P}(k), \Gamma)$, $l = 1, 2, 3$, $\mathbf{v}(k)$ may not necessarily be a descent direction. In such cases, the cost function will not decrease along that direction, and the resulting step size $\alpha(k)$ from the backtracking algorithm will be approximately zero. This is the reason for considering the second decision-making block of Fig. 4.1. When such a condition is encountered, one needs to terminate the inner optimization loop and update the value of μ such that

$$\tilde{f}_l(\mathbf{P}(k) + \alpha(k')\mathbf{v}(k'), \Gamma') - \tilde{f}_l(\mathbf{P}(k), \Gamma') < 0,$$

for some $\alpha(k') \in (0, \alpha_{max}]$, where α_{max} is the maximum value taken by the step size. This procedure will enable one to have a descent direction at iteration k'. However, the nonlinear nature of $\tilde{f}_l$, $l = 1, 2, 3$, is a barrier to achieving this. As a remedy, according to Algorithm 4.1 and Fig. 4.1, the inner optimization loop is terminated once the step size is approximately zero, and the value of μ is updated to be any $\mu \in \mathbb{R}_{>1}$. Note that the parameter μ is used to update Γ, which is directly related to the termination condition of the outer optimization loop (as seen in Algorithm 4.1). Using large values for this parameter may lead to the premature termination of the optimization algorithm. As explained in the following remark, the considered μ will make the value of $\langle \mathbf{v}(k'), \mathbf{g}(k') \rangle$ more negative at iteration k', increasing the likelihood of having a descent direction.

Remark 4.3. By expanding $\langle \mathbf{v}(k'), \mathbf{g}(k') \rangle$ for $l = 2, 3$, it can be easily seen that any $\mu \in \mathbb{R}_{>1}$ will make the inner product more negative. For $l = 1$, however, this is not necessarily the case. If $\nabla\tilde{\lambda}^1_{ij}(\mathbf{P}(k)) \cdot \nabla\tilde{\lambda}^2_{ij}(\mathbf{P}(k))$ is negative, a $\mu \in \mathbb{R}_{>1}$ will make (4.25) more positive, which is contrary to the present objective. However, note that $\nabla\tilde{\lambda}^1_{ij}(\mathbf{P}(k)) \cdot \nabla\tilde{\lambda}^2_{ij}(\mathbf{P}(k)) \leq 0$ indicates that the algorithm has reached a local or global optimum, and therefore, no further move is required in any direction.

In order to determine the step size $\alpha(k)$ in (4.12) via the backtracking line search, as long as the Armijo-Goldstein condition [14] given by

$$\tilde{f}_l(\mathbf{P}(k) + \alpha(k)\mathbf{v}(k), \Gamma) \leq \tilde{f}_l(\mathbf{P}(k), \Gamma) + \nu\alpha(k)\langle \mathbf{v}(k), \mathbf{g}(k)\rangle \tag{4.28}$$

is not satisfied, the step size will be shrinked by a factor of $\theta \in (0, 1)$ (note that $\nu \in (0, 1)$ as well in the above inequality, which determines the expected amount of decrease in the cost function). This procedure also needs to be implemented in a distributed way. Estimating the global variable $\tilde{f}_l$ for $l = 1, 2, 3$, using the methods of [4] or [33] would be computationally demanding and time consuming, which is not desirable given the nodes' limited resources. This problem is addressed in the following lemmas by finding bounds for the Armijo-Goldstein condition.

Lemma 4.3. *For $l = 1$ in the optimization problem (4.7), the step size $\alpha(k)$ is reduced by a factor $\theta \in (0, 1)$ until the following condition is satisfied*

$$\alpha(k) \cdot L \cdot \|\boldsymbol{v}(k)\|_F + \Gamma^{-1}\big(I_1(\boldsymbol{P}(k) + \alpha(k)\boldsymbol{v}(k)) - I_1(\boldsymbol{P}(k))\big) - \nu\alpha(k)\langle \boldsymbol{v}(k), \boldsymbol{g}(k)\rangle \leq 0, \tag{4.29}$$

where L is the Lipschitz constant of the GAC, introducing an upper bound on the magnitude of the derivatives of the GAC.

Proof. By expanding the function $\tilde{f}_l$ for $l = 1$, the Armijo-Goldstein condition (4.28) is re-written as

$$
\begin{aligned}
-\,\tilde{\lambda}(\mathbf{P}(k)+\alpha(k)\mathbf{v}(k))+\tilde{\lambda}(\mathbf{P}(k))+\Gamma^{-1}\big(I_1(\mathbf{P}(k)+\alpha(k)\mathbf{v}(k))-I_1(\mathbf{P}(k))\big)-& \\
\nu\alpha(k)\langle\mathbf{v}(k),\mathbf{g}(k)\rangle\leq 0.& \quad (4.30)
\end{aligned}
$$

If (4.30) was to be implemented in its present form, the first two GAC terms would need to be estimated for each step size value before possibly shrinking it. A better approach is to approximate these terms in order to avoid further complication. Since the GAC is a locally Lipschitz function with constant L, one can write

$$
|\tilde{\lambda}(\mathbf{P}(k)+\alpha(k)\mathbf{v}(k))-\tilde{\lambda}(\mathbf{P}(k))|\leq\alpha(k)\cdot L\cdot\|\mathbf{v}(k)\|_{\mathrm{F}}. \tag{4.31}
$$

By combining (4.30) and (4.31), one arrives at (4.29). This completes the proof. ■

Remark 4.4. In deriving Lemma 4.3, the term $-\tilde{\lambda}(\mathbf{P}(k)+\alpha(k)\mathbf{v}(k))+\tilde{\lambda}(\mathbf{P}(k))$, which may have a negative value, is approximated by an always positive term. At any iteration k, the Armijo-Goldstein condition for $l=1$ given by (4.30) may be satisfied, whereas the approximated value (4.29) may still be positive. The result is that at each iteration k, the step size obtained via the approximate backtracking line search will be smaller compared to the case when the global values are used, i.e., the approximate method will take smaller steps at each iteration toward the optimum. The choice of the real constant L determines how close the step sizes obtained from the approximate and exact methods are. It is possible that for some applications, depending on the magnitude of the terms in (4.29), the terminal condition of Lemma 4.3 may not be satisfied, in which case the approximate method cannot be used. In such a case, at the expense of longer computational time, the exact values of the GAC may be used in (4.30) to obtain the step sizes.

Lemma 4.4. *For $l = 2, 3$ in the optimization problem (4.7), the step size $\alpha(k)$ is reduced by the factor $\theta \in (0, 1)$ at consecutive steps until the following inequality is satisfied*

$$h(k) - \Gamma^{-1} \log(\frac{\alpha(k) \cdot L \cdot \|\boldsymbol{v}(k)\|_F}{\tilde{\lambda}(\boldsymbol{P}(k)) - \underline{\lambda}} + 1) \leq 0, \tag{4.32}$$

where, for $l = 2$,

$$h(k) = \alpha(k) \sum_i \sum_j v_{ij}(k) + Q(k), \tag{4.33}$$

and for $l = 3$,

$$h(k) = -T_{sys}(\boldsymbol{P}(k) + \alpha(k)\boldsymbol{v}(k)) + T_{sys}(\boldsymbol{P}(k)) + Q(k), \tag{4.34}$$

and

$$Q(k) = -\nu\alpha(k)\langle \boldsymbol{v}(k), \boldsymbol{g}(k)\rangle + \Gamma^{-1}\Big(\sum_i \sum_j \log\big(\frac{P_{ij}^{up} - P_{ij}(k)}{P_{ij}^{up} - P_{ij}(k) - \alpha(k)v_{ij}(k)}\big) + \sum_i \sum_j \log\big(\frac{P_{ij}(k) - P_{ij}^{low}}{P_{ij}(k) + \alpha(k)v_{ij}(k) - P_{ij}^{low}}\big)\Big), \tag{4.35}$$

for all $i \in \mathbb{N}_n$ *and* $j \in N_i^{in}$.

Proof. By expanding $\tilde{f}_2$ and $\tilde{f}_3$, the Armijo-Goldstein condition (4.28) becomes

$$h(k) - \Gamma^{-1} \log \big(\frac{\tilde{\lambda}(\mathbf{P}(k) + \alpha(k)\mathbf{v}(k)) - \underline{\lambda}}{\tilde{\lambda}(\mathbf{P}(k)) - \underline{\lambda}} \big) \leq 0, \tag{4.36}$$

where $h(k)$ is given by (4.33) or (4.34). The last term in the left-hand side of (4.36) is approximated to make the distributed algorithm less computationally demanding. Using the Lipschitz property of the GAC again, (4.31) can be re-written as

$$\frac{\tilde{\lambda}(\mathbf{P}(k) + \alpha(k)\mathbf{v}(k)) - \underline{\lambda}}{\tilde{\lambda}(\mathbf{P}(k)) - \underline{\lambda}} \geq \frac{\alpha(k) \cdot L \cdot \|\mathbf{v}(k)\|_{\mathrm{F}}}{\tilde{\lambda}(\mathbf{P}(k)) - \underline{\lambda}} + 1. \tag{4.37}$$

As per Assumption 4.2, the initial transmission power matrix is chosen to be strictly feasible, and consequently, the denominator in (4.37) is not equal to zero. Manipulating (4.37), one arrives at

$$h(k) - \Gamma^{-1} \log \big(\frac{\tilde{\lambda}(\mathbf{P}(k) + \alpha(k)\mathbf{v}(k)) - \underline{\lambda}}{\tilde{\lambda}(\mathbf{P}(k)) - \underline{\lambda}} \big) \leq h(k) - \Gamma^{-1} \log \big(\frac{\alpha(k) \cdot L \cdot \|\mathbf{v}(k)\|_{\mathrm{F}}}{\tilde{\lambda}(\mathbf{P}(k)) - \underline{\lambda}} + 1 \big). \tag{4.38}$$

As the left-hand side of (4.38) is equal to (4.36), the Armijo-Goldstein condition is guaranteed to be satisfied if the right-hand side of (4.38) becomes less than or equal to zero. This completes the proof. ∎

Similar to Lemma 4.3, it is shown in Lemma 4.4 that reducing the step size properly at each iteration ensures that the Armijo-Goldstein condition is eventually satisfied.

However, the issue with Lemma 4.3 not being satisfied in some applications (as mentioned in Remark 4.4) is less pronounced for Lemma 4.4. The reason is that even if the approximate term has a large positive value, taking its logarithm as in (4.32) will reduce it. Nonetheless, if the same issue comes up in Lemma 4.4, the global value of the GAC could be used to obtain the step size at the expense of additional computational time. Distributed implementations of Lemmas 4.3 and 4.4 are straightforward, as nodes only need to use a basic consensus protocol to determine the required values (there is no need to estimate new global values such as the GAC anymore). Additionally, it is assumed that each node knows an upper bound of the GAC function's Lipschitz constant L.

Remark 4.5. The locally Lipschitz property of the GAC function, used in the proofs of Lemmas 4.3 and 4.4, holds only in the neighborhood of the original point ($\mathbf{P}(k)$ in these cases). In using the approximate backtracking approaches, if $\alpha(k)$ is too large, (4.31) will not necessarily hold. By appropriately choosing α_{max}, this problem can be avoided.

Having explained all the components in the Fig. 4.1, the proposed distributed optimization procedure is given in Algorithm 4.3. The parameters of this algorithm which are not previously described are μ_0, β_0, ζ and m_{max}. The parameter μ_0 is the initial value of μ, which is updated as the algorithm proceeds. The parameter β_0 is a real constant used to calculate $\beta(k)$, which is the step size used to numerically calculate the partial derivatives of the GAC. The parameter ζ in line 19 of Algorithm 4.3 determines the threshold below which the step size can be considered to be approximately zero. Ultimately, the parameter m_{max} is the maximum number of iterations the inner optimization loop runs.

Before starting the approximate backtracking line search in Algorithm 4.3 (line 16), an additional step is considered for reducing the step size to ensure that moving along the determined search direction will not take the algorithm out of the corresponding feasible region $\mathcal{R}_l$; otherwise, certain logarithmic terms in the formulations of Lemmas 4.3 and 4.4 would be undefined. The implementation of this step is presented next.

Remark 4.6. In determining the maximum value of $\alpha(k)$ which ensures that $\mathbf{P}(k) + \alpha(k) \cdot \mathbf{v}(k) \in \mathcal{R}_l$ in Algorithm 4.3, the nodes need to first implement the local constraints $h^2_{ij}(k)$ and $h^3_{ij}(k)$, for all $i, j \in \mathbb{N}_n$ and $k \in \mathbb{N}$. That is, each node should check whether $P_{ij}(k) + \alpha(k)v_{ij}(k)$ is within the range $[P^{\text{low}}_{ij}, P^{\text{up}}_{ij}]$. Once each node finds the maximum $\alpha(k)$ for its transmission power levels, they can communicate with one another, share their values of the step size, and choose the minimum value among the step sizes of each node in that iteration to move onto the global constraint. Unlike the backtracking line search which was approximated to make it computationally more efficient, the nodes next need to determine the exact values of the required global variables, such as the GAC. To check the global constraint $h^1_l(k)$ for $l = 1$, the nodes can use a basic consensus protocol to determine the value of $\alpha(k)$ which satisfies the following condition

$$\bar{\mathrm{P}} - \sum_i \sum_j \big(P_{ij}(k) + \alpha(k)v_{ij}(k)\big) \geq 0.$$

On the other hand, to check the global constraint $h_l^1(k)$ for $l = 2, 3$, the nodes need to use the method of [1] or [33] to determine the maximum value of $\alpha(k)$ that satisfies the following inequality

$$\tilde{\lambda}(\mathbf{P}(k) + \alpha(k)\mathbf{v}(k)) - \underline{\lambda} \geq 0.$$

As far as the scalability and run-time of the proposed procedure is concerned, the main difference between Algorithm 4.3 and its centralized counterpart is the estimation of the GAC using local information and the additional computation time it requires. Other than that, the two algorithms (centralized and distributed) are almost identical. Whichever method is utilized to estimate the GAC in a distributed manner and whatever its computational complexity may be, it is repeated at most $2(n^2 - n) + 1$ times. Also, given the GAC constraint for $l = 2, 3$ in the optimization problem (4.7) and the need to use a GAC estimation algorithm in line 13-15 of Algorithm 4.3 (see Remark 4.6), the selected estimation algorithm will need to be executed $\lceil \log(\frac{\zeta}{\alpha_{max}})/\log(\theta) \rceil$ times in the worst case scenario. Note that this value is not dependent on the size of the network.

4.4 Convergence analysis of the optimization algorithm

The asymptotic convergence of the proposed optimization algorithms to a local or global minimum of the constrained optimization problems P1-P3 is investigated next.

Algorithm 4.3. Proposed distributed optimization algorithm.

Given strictly feasible $\mathbf{P} = \mathbf{P}^0 \in \mathbb{R}^{n \times n}$, initialize $\Gamma = \Gamma^0 \in \mathbb{R}_{>0}$ and $k = 1$.
Choose arbitrary constants $\nu \in (0,1)$, $\theta \in (0,1)$, $\mu_0 \in \mathbb{R}_{>1}$, β_0, ϵ, ζ, $\alpha_{max} \in \mathbb{R}_{>0}$, $m_{max} \in \mathbb{N}$ and consider the prescribed parameters $\mathbf{P}^{\text{up}}$, $\mathbf{P}^{\text{low}} \in \mathbb{R}^{n \times n}$, and $\underline{\lambda}$ or $\bar{\mathrm{P}} \in \mathbb{R}_{>0}$.
while $m\Gamma^{-1} > \epsilon$ **do**
$\mu = \mu_0$
  for $m_{iter} = 1 : m_{max}$ **do**
    Compute $\mathbf{v}(k)$ according to (4.13)-(4.15).
    Use Lemma 4.1 and Algorithm 4.2 to determine whether $\mathbf{v}(k)$ is not a descent direction.
    if not a descent direction **do**
      $\mu = \max(\mu_0, \mu')$, where μ' is a value obtained from Lemma 4.2.
      break
    $\alpha(k) = \alpha_{max}$
    while $\mathbf{P}(k) + \alpha(k) \cdot \mathbf{v}(k) \notin \mathcal{R}_l$ according to Remark 4.6 **do**
      $\alpha(k) = \theta \cdot \alpha(k)$
    end while
    while the inequalities in Lemmas 4.3 or 4.4 are not satisfied **do**
      $\alpha(k) = \theta \cdot \alpha(k)$
    end while
    if $\alpha(k) \leq \zeta$ **do**
      $\mu = \mu_0$
      break
    $\mathbf{P}(k+1) = \mathbf{P}(k) + \tau(k) \cdot \mathbf{v}(k)$
    $k = k + 1$
  end
  $\Gamma = \mu\Gamma$
end while

Remark 4.7. Consider an asymmetric network composed of n nodes represented by a weighted digraph. Let Assumptions 4.1, 4.2 and 4.3 hold. Using the interior point algorithm, the following relations hold for $l = 1, 2, 3$

1. $\lim_{k\to\infty} f_l(\mathbf{P}(k)) = v_l^*$,

2. $\lim_{k\to\infty} \tilde{f}_l(\mathbf{P}(k), \Gamma) = v_l^*$,

3. $\lim_{k\to\infty} \Gamma^{-1} \cdot I_l(\mathbf{P}(k)) = 0$,

4. $\lim_{k\to\infty} \mathbf{P}(k) = \mathbf{P}_l^*$,

where v_l^*, for $l = 1$, is a local minimum of the optimization problem P1, and for $l = 2, 3$, is the global minimum of the optimization problems P2 and P3. $\mathbf{P}_l^*$ is the corresponding minimizer of v_l^* [13,28].

The relations in Remark 4.7 are presented in [28,29], and are repeated here for ease of reference. The remark states that sequences $\mathbf{P}(k)$ converging to local or global minima of the problems P1-P3 exist. Now, one needs to show that the subgradient method with the approximate backtracking line search can generate such sequences. To this end, for each Γ, the unique global minimum of optimization problem (4.7), for $l = 1, 2, 3$, is denoted by $\mathbf{P}_\Gamma^*$. These global minima are called central points of the interior point algorithm [14]. Additionally, let $\mathbf{P}_\Gamma^0$ denote the power matrix that the inner optimization loop starts with (when $m_{iter} = 1$), for any $\Gamma \in \mathbb{R}_{>0}$. Starting from $\mathbf{P}_\Gamma^0$, $\tilde{f}_l(\mathbf{P}(k), \Gamma)$ must converge to $\tilde{f}_l(\mathbf{P}_\Gamma^*, \Gamma)$, for $l = 1, 2, 3$, using the subgradient method with approximate backtracking line search. Since $\tilde{f}_l(\mathrm{P}_{\Gamma}^*, \Gamma) \to v_l^*$ as $k \to \infty$ [14], $\tilde{f}_l(\mathbf{P}(k), \Gamma)$ is guaranteed to converge to v_l^*.

Theorem 4.1. *Consider an asymmetric network composed of n nodes represented by a weighted digraph, as described earlier. Using the interior point method in conjunction with the subgradient approach and the backtracking line search to solve the constrained optimization problems P1-P3, as $k \to \infty$, the transmission power matrix $\boldsymbol{P}(k)$ asymptotically converges to a stationary matrix $\boldsymbol{P}_l^* \in \mathcal{R}_l$ corresponding to a local minimum v_1^* and to the global minima v_2^* and v_3^* of the optimization problem (4.7).*

Proof. Given $\mathbf{g}(k) \in \partial \tilde{f}_l(\mathbf{P}(k), \Gamma)$, for any $k \in \mathbb{N}$, $l \in \mathbb{N}_3$ and $\Gamma \in \mathbb{R}_{>0}$, the search direction $\mathbf{v}(k)$ is a descent direction as per the previously discussed lemmas and remarks. As a result, the backtracking line search algorithm will eventually stop for some $\alpha(k) \in (0, \alpha_{max}]$. Then

$$\tilde{f}_l(\mathbf{P}(k), \Gamma) - \tilde{f}_l(\mathbf{P}(k-1), \Gamma) \leq \nu \cdot \alpha(k-1) \cdot \langle \mathbf{v}(k-1), \mathbf{g}(k-1) \rangle. \tag{4.39}$$

Given that $\langle \mathbf{v}(k), \mathbf{g}(k) \rangle < 0$, for any $k \in \mathbb{N}$, (4.39) can be rewritten as

$$\tilde{f}_l(\mathbf{P}(k), \Gamma) - \tilde{f}_l(\mathbf{P}(k-1), \Gamma) \leq 0. \tag{4.40}$$

From the definition of subgradient (Definition 4.1), any $\mathbf{g}(k) \in \partial \tilde{f}_l(\mathbf{P}(k), \Gamma)$, including the one corresponding to the search direction (i.e. $\mathbf{v}(k) = -\mathbf{g}(k)$), satisfies the following inequality

$$\tilde{f}_l(\mathbf{P}(k-1),\Gamma) + \langle \mathbf{P}(k-1) - \mathbf{P}^*_\Gamma, \mathbf{g}(k-1)\rangle \leq \tilde{f}_l(\mathbf{P}^*_\Gamma,\Gamma). \tag{4.41}$$

Combining (4.40) and (4.41) yields

$$\tilde{f}_l(\mathbf{P}(k),\Gamma) - \tilde{f}_l(\mathbf{P}^*_\Gamma,\Gamma) \leq \langle \mathbf{P}(k-1) - \mathbf{P}^*_\Gamma, \mathbf{g}(k-1)\rangle. \tag{4.42}$$

The right-hand side of (4.42) can then be rewritten as

$$\frac{1}{\alpha(k-1)}\langle \mathbf{P}(k-1) - \mathbf{P}^*_\Gamma, \alpha(k-1)\mathbf{g}(k-1)\rangle = \frac{1}{\alpha(k-1)}\langle \mathbf{P}(k-1) - \mathbf{P}^*_\Gamma, \mathbf{P}(k) - \mathbf{P}(k-1)\rangle. \tag{4.43}$$

By manipulating the second term in the inner product of the right-hand side of (4.43), (4.42) becomes

$$\begin{aligned}\tilde{f}_l(\mathbf{P}(k),\Gamma) - \tilde{f}_l(\mathbf{P}^*_\Gamma,\Gamma) \leq \frac{-1}{\alpha(k-1)}&\langle \mathbf{P}(k-1) - \mathbf{P}^*_\Gamma, \mathbf{P}(k-1) - \mathbf{P}^*_\Gamma\rangle \\ &+ \frac{1}{\alpha(k-1)}\langle \mathbf{P}(k-1) - \mathbf{P}^*_\Gamma, \mathbf{P}(k) - \mathbf{P}^*_\Gamma\rangle,\end{aligned} \tag{4.44}$$

and after further manipulations, one arrives at

$$\begin{aligned}\tilde{f}_l(\mathbf{P}(k),\Gamma) - \tilde{f}_l(\mathbf{P}^*_\Gamma,\Gamma) \leq \frac{-1}{\alpha(k-1)}&\langle \mathbf{P}(k-1) - \mathbf{P}^*_\Gamma, \mathbf{P}(k-1) - \mathbf{P}^*_\Gamma\rangle + \\ \frac{1}{\alpha(k-1)}\langle \mathbf{P}(k) - \mathbf{P}^*_\Gamma, \mathbf{P}(k) - \mathbf{P}^*_\Gamma\rangle &+ \frac{1}{\alpha(k-1)}\langle \mathbf{P}(k-1) - \mathbf{P}(k), \mathbf{P}(k) - \mathbf{P}^*_\Gamma\rangle.\end{aligned} \tag{4.45}$$

The third term in the right-hand side of the above inequality is rewritten to obtain

$$\tilde{f}_l(\mathbf{P}(k),\Gamma)-\tilde{f}_l(\mathbf{P}^*_\Gamma,\Gamma)\leq\frac{-1}{\alpha(k-1)}\langle\mathbf{P}(k-1)-\mathbf{P}^*_\Gamma,\mathbf{P}(k-1)-\mathbf{P}^*_\Gamma\rangle+$$
$$\frac{1}{\alpha(k-1)}\langle\mathbf{P}(k)-\mathbf{P}^*_\Gamma,\mathbf{P}(k)-\mathbf{P}^*_\Gamma\rangle+\langle\mathbf{g}(k-1),\mathbf{P}^*_\Gamma-\mathbf{P}(k)\rangle. \quad (4.46)$$

The right-hand side of (4.42) is re-expressed as

$$\tilde{f}_l(\mathbf{P}^*_\Gamma,\Gamma)-\tilde{f}_l(\mathbf{P}(k-1),\Gamma)\geq\langle\mathbf{g}(k-1),\mathbf{P}^*_\Gamma-\mathbf{P}(k)\rangle+\alpha(k)\langle\mathbf{g}(k-1),\mathbf{g}(k-1)\rangle. \quad (4.47)$$

Now, combining (4.46) and (4.47) yields

$$\tilde{f}_l(\mathbf{P}(k),\Gamma)+\tilde{f}_l(\mathbf{P}(k-1),\Gamma)-2\tilde{f}_l(\mathbf{P}^*_\Gamma,\Gamma)\leq\frac{-1}{\alpha(k-1)}\langle\mathbf{P}(k-1)-\mathbf{P}^*_\Gamma,\mathbf{P}(k-1)-\mathbf{P}^*_\Gamma\rangle+$$
$$\frac{1}{\alpha(k-1)}\langle\mathbf{P}(k)-\mathbf{P}^*_\Gamma,\mathbf{P}(k)-\mathbf{P}^*_\Gamma\rangle-\alpha(k)\langle\mathbf{g}(k-1),\mathbf{g}(k-1)\rangle. \quad (4.48)$$

Note that $\langle\mathbf{g}(k-1),\mathbf{g}(k-1)\rangle$ in the right-hand side of (4.48) is non-negative. In addition, since $\tilde{f}_l(\mathbf{P}^*_\Gamma,\Gamma)$ is the minimum of the cost function $\tilde{f}_l$, $l=1,2,3$, for any $\Gamma\in\mathbb{R}_{>0}$, it is evident that $\tilde{f}_l(\mathbf{P}(k-1),\Gamma)-\tilde{f}_l(\mathbf{P}^*_\Gamma,\Gamma)\geq 0$. Hence, (4.48) can be rewritten as

$$\tilde{f}_l(\mathbf{P}(k),\Gamma)-\tilde{f}_l(\mathbf{P}^*_\Gamma,\Gamma)\leq\frac{1}{\alpha(k-1)}\langle\mathbf{P}(k)-\mathbf{P}^*_\Gamma,\mathbf{P}(k)-\mathbf{P}^*_\Gamma\rangle$$
$$-\frac{1}{\alpha(k-1)}\langle\mathbf{P}(k-1)-\mathbf{P}^*_\Gamma,\mathbf{P}(k-1)-\mathbf{P}^*_\Gamma\rangle. \quad (4.49)$$

Averaging both sides of (4.49) yields

$$\frac{1}{k}\sum_{j=2}^{k}\big(\tilde{f}_l(\mathbf{P}(j),\Gamma)-\tilde{f}_l(\mathbf{P}_\Gamma^*,\Gamma)\big)\le\frac{1}{k}\sum_{j=2}^{k}\Big(\frac{1}{\alpha(j-1)}\langle\mathbf{P}(j)-\mathbf{P}_\Gamma^*,\mathbf{P}(j)-\mathbf{P}_\Gamma^*\rangle - \frac{1}{\alpha(j-1)}\langle\mathbf{P}(j-1)-\mathbf{P}_\Gamma^*,\mathbf{P}(j-1)-\mathbf{P}_\Gamma^*\rangle\Big), \quad (4.50)$$

which can then be rewritten as

$$\frac{1}{k}\sum_{j=2}^{k}\big(\tilde{f}_l(\mathbf{P}(j),\Gamma)-\tilde{f}_l(\mathbf{P}_\Gamma^*,\Gamma)\big)\le\frac{1}{k}\cdot\frac{1}{\min\alpha(j-1)|_{j=2}^{k}}\cdot\sum_{j=2}^{k}\Big(\langle\mathbf{P}(j)-\mathbf{P}_\Gamma^*,\mathbf{P}(j)-\mathbf{P}_\Gamma^*\rangle - \langle\mathbf{P}(j-1)-\mathbf{P}_\Gamma^*,\mathbf{P}(j-1)-\mathbf{P}_\Gamma^*\rangle\Big). \quad (4.51)$$

The right-hand side of (4.51) further simplifies to

$$\frac{1}{k}\sum_{j=2}^{k}\big(\tilde{f}_l(\mathbf{P}(j),\Gamma)-\tilde{f}_l(\mathbf{P}_\Gamma^*,\Gamma)\big)\le\frac{1}{k}\cdot\frac{1}{\min\alpha(j-1)|_{j=2}^{k}}\cdot\Big(\langle\mathbf{P}(k)-\mathbf{P}_\Gamma^*,\mathbf{P}(k)-\mathbf{P}_\Gamma^*\rangle - \langle\mathbf{P}(1)-\mathbf{P}_\Gamma^*,\mathbf{P}(1)-\mathbf{P}_\Gamma^*\rangle\Big). \quad (4.52)$$

Note that both inner products in the right-hand side of (4.52) are non-negative. Also, given the definition of the central points $\mathbf{P}_\Gamma^*$, one has

$$\tilde{f}_l(\mathbf{P}(k),\Gamma)-\tilde{f}_l(\mathbf{P}_\Gamma^*,\Gamma)\le\frac{1}{k}\sum_{j=2}^{k}\big(\tilde{f}_l(\mathbf{P}(j),\Gamma)-\tilde{f}_l(\mathbf{P}_\Gamma^*,\Gamma)\big). \quad (4.53)$$

Now, (4.52) is transformed to

$$\tilde{f}_l(\mathbf{P}(k), \Gamma) - \tilde{f}_l(\mathbf{P}^*_\Gamma, \Gamma) \leq \frac{1}{k} \cdot \frac{1}{\min \alpha(j-1)|_{j=2}^{k}} \cdot \langle \mathbf{P}(k) - \mathbf{P}^*_\Gamma, \mathbf{P}(k) - \mathbf{P}^*_\Gamma \rangle. \qquad (4.54)$$

According to Algorithm 4.3, every time the inner optimization loop is terminated, Γ increases and the values of $\tilde{f}_l(\mathbf{P}^*_\Gamma, \Gamma)$ and $\mathbf{P}^*_\Gamma$ are updated. As $k \to \infty$, the right-hand side of (4.54) approaches zero, and also $\tilde{f}_l(\mathrm{P}^*_\Gamma, \Gamma) \to v^*_l$ (see Remark 4.7). This means that as $k \to \infty$, $\tilde{f}_l(\mathbf{P}(k), \Gamma)$ converges to v^*_l. Also, according to Remark 4.7, $\mathbf{P}(k)$ approaches $\mathbf{P}^*_l$ as k increases. This completes the proof. ■

Note that according to Theorem 4.1, Algorithm 4.3 converges to a neighborhood of the solutions of the optimization problems P1-P3. At the cost of a longer convergence time, this neighborhood can be made arbitrarily small by a proper choice of the parameter ϵ in the termination condition of Algorithm 4.3 (line 3).

Remark 4.8. If the elements of the transmission power matrix converge to the boundaries of the set $\mathcal{R}_l$, for any $l \in \mathbb{N}_3$, the corresponding terms of $\mathbf{g}(k)$ will go to infinity. Because the initial transmission power matrix is chosen to be strictly inside the feasible set $\mathcal{R}_l$, its elements will only converge to the boundaries as $k \to \infty$ if an optimum lies on the boundaries. In other words, $\mathbf{g}(k)$ becoming infinity implies that the algorithm has already reached a solution.

4.5 Simulation Results

The results of the centralized optimization algorithm (with the exact GAC and backtracking line search values) concerning the three variants of the optimization algorithm (4.7) are presented first. It is then followed by the results of the distributed optimization algorithm proposed in this study (with the estimated GAC and approximate backtracking line search values). The results are also compared with the outcome of the *fmincon* function of MATLAB®. Note that the centralized algorithms for $l = 2, 3$ are available in [28] and [29], respectively, and can be derived in a similar fashion for $l = 1$.

Example 4.1. To investigate the efficacy of Algorithm 4.3, the four-node experimental asymmetric network of [15] is examined here. It is assumed that the communication link from node 2 to node 4 is not feasible in this network due to environmental constraints, i.e., $P_{42}(k)$ is equal to zero for all $k \in \mathbb{N}$ and is not considered in the optimization algorithm. Assuming that for every $i \in \mathbb{N}_n$ and $j \in N_i^{in}$, $P_{ij}^{\text{low}} = 1$ (the smallest power level required to establish a link) and $P_{ij}^{\text{up}} = 4$, the initial transmission power matrix is chosen as

$$\mathbf{P}^0 = \begin{bmatrix} 0 & 1.3 & 1.5 & 1.6 \\ 1.2 & 0 & 1.5 & 1.3 \\ 1.7 & 1.7 & 0 & 1.4 \\ 1.5 & 0 & 1.5 & 0 \end{bmatrix}.$$

Using (4.4), the resulting initial weight matrix of the corresponding digraph is as

$$\mathbf{W} = \begin{bmatrix} 0 & 0.4474 & 0.5889 & 0.4846 \\ 0.5178 & 0 & 0.5932 & 0.4653 \\ 0.5214 & 0.5682 & 0 & 0.4677 \\ 0.6192 & 0 & 0.5224 & 0 \end{bmatrix},$$

where the network's initial GAC is $\tilde{\lambda}(\mathbf{P}^0) = 1.6027$ and initial total transmission power is $\sum_i \sum_j P_{ij}^0 = 16.2$. Assuming

$$\mathbf{q} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}, \ \mathbf{K} = \begin{bmatrix} 0 & 2 & 1 & 2 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 1 \\ 1 & 0 & 2 & 0 \end{bmatrix}, \ \tau = \begin{bmatrix} 0 & 1.2 & 1.5 & 2 \\ 1.1 & 0 & 1.8 & 1.4 \\ 1.5 & 1.7 & 0 & 0.8 \\ 1.2 & 0 & 0.9 & 0 \end{bmatrix},$$

$\mathfrak{e}_i^0 = 200$ and $\mathfrak{e}_{ij}^r = 1.5$ for every $i \in \mathbb{N}_n$ and $j \in N_i^{in}$, the initial lifetime of nodes are

$$T(\mathbf{P}^0) = [12.7226, 16.6528, 14.6520, 12.7714]^{\mathrm{T}}.$$

Furthermore, it is required that the GAC be greater than or equal to $\underline{\lambda} = 1.5$ and the total transmission power be less than or equal to $\bar{P} = 20$. As a result, the initial transmission power matrix is strictly inside the feasible set $\mathcal{R}_l$, for all $l \in \mathbb{N}_3$.

4.5.1 Minimizing $\tilde{f}_1$ in (4.7)

To implement the centralized optimization algorithm for $l = 1$ in the optimization problem (4.7), i.e., maximizing the GAC with constraints on the total transmission power and each transmission level, the maximum number of iterations for the inner optimization loop is chosen to be $m_{max} = 30$. The design parameters of the backtracking line search are selected to be $\nu = 0.01$, $\theta = 0.95$, and $\alpha_{max} = 1$. Additionally, the parameter used to numerically calculate the supergradients of the GAC is $\beta = 0.1$, and the coefficient by which Γ, the weight given to the penalty terms $I_1(\mathbf{P}(k))$, is multiplied with at the end of an inner optimization loop is at least equal to $\mu_0 = 5$. The initial value of Γ is $\Gamma^0 = 150$. The performance of the algorithm is evaluated by choosing $\epsilon = 10^{-4}$ as the termination condition of the outer optimization loop. The parameter ζ in line 19 of Algorithm 4.3 is set equal to 10^{-4}. Finally, the search direction is chosen as $\mathbf{v}(k) = -\mathbf{g}^{-}(k)$ for all $k \in \mathbb{N}$. Using the above-mentioned parameters, the optimal transmission power matrix is obtained as

$$\mathbf{P}^*_{c,1} = \begin{bmatrix} 0 & 1.1922 & 1.8520 & 1.8751 \\ 1.3183 & 0 & 1.4881 & 2.6007 \\ 1.6812 & 1.6015 & 0 & 1.9287 \\ 2.6221 & 0 & 1.8401 & 0 \end{bmatrix},$$

resulting in the following weight matrix

$$\mathbf{W}_{c,1}^{*} = \begin{bmatrix} 0 & 0.4337 & 0.7409 & 0.5893 \\ 0.5365 & 0 & 0.5898 & 0.6462 \\ 0.5150 & 0.5387 & 0 & 0.6468 \\ 0.8646 & 0 & 0.5629 & 0 \end{bmatrix}.$$

The corresponding network GAC is $\tilde{\lambda}(\mathbf{P}_{c,1}^{*}) = 2.2210$ and the total transmission power is $\sum_{i\in\mathbb{N}_n}\sum_{j\in\mathbb{N}_n} {P_{ij}}_{c,1}^{*} = 20$. The evolution of the GAC and the total transmission power of the network as the iteration index k increases is shown in Fig. 4.2, and the evolution of the individual transmission powers of each node is presented in Fig. 4.3.

As mentioned earlier, the algorithm of [33], proposed for obtaining all the eigenvalues of the graph Laplacian, is utilized to obtain the *exact* value of the eigenvalue corresponding to the GAC in a distributed manner. Moreover, Lemma 4.3 is used to obtain the step sizes; however, the problem pointed out in Remark 4.4 arises in this example. Considering an upper bound of $L = 1.5$ for the Lipschitz constant of the GAC function, regardless of the values of the design parameters, the condition of Lemma 4.3 is not satisfied after a few iterations of the optimization algorithm. The reason is that as seen in (4.29), only the second term can have negative values, and only while its coefficient Γ is not too large, the inequality of Lemma 4.3 may be satisfied. However, as the value of Γ increases as per line 25 of Algorithm 4.3, inequality (4.29) is not satisfied

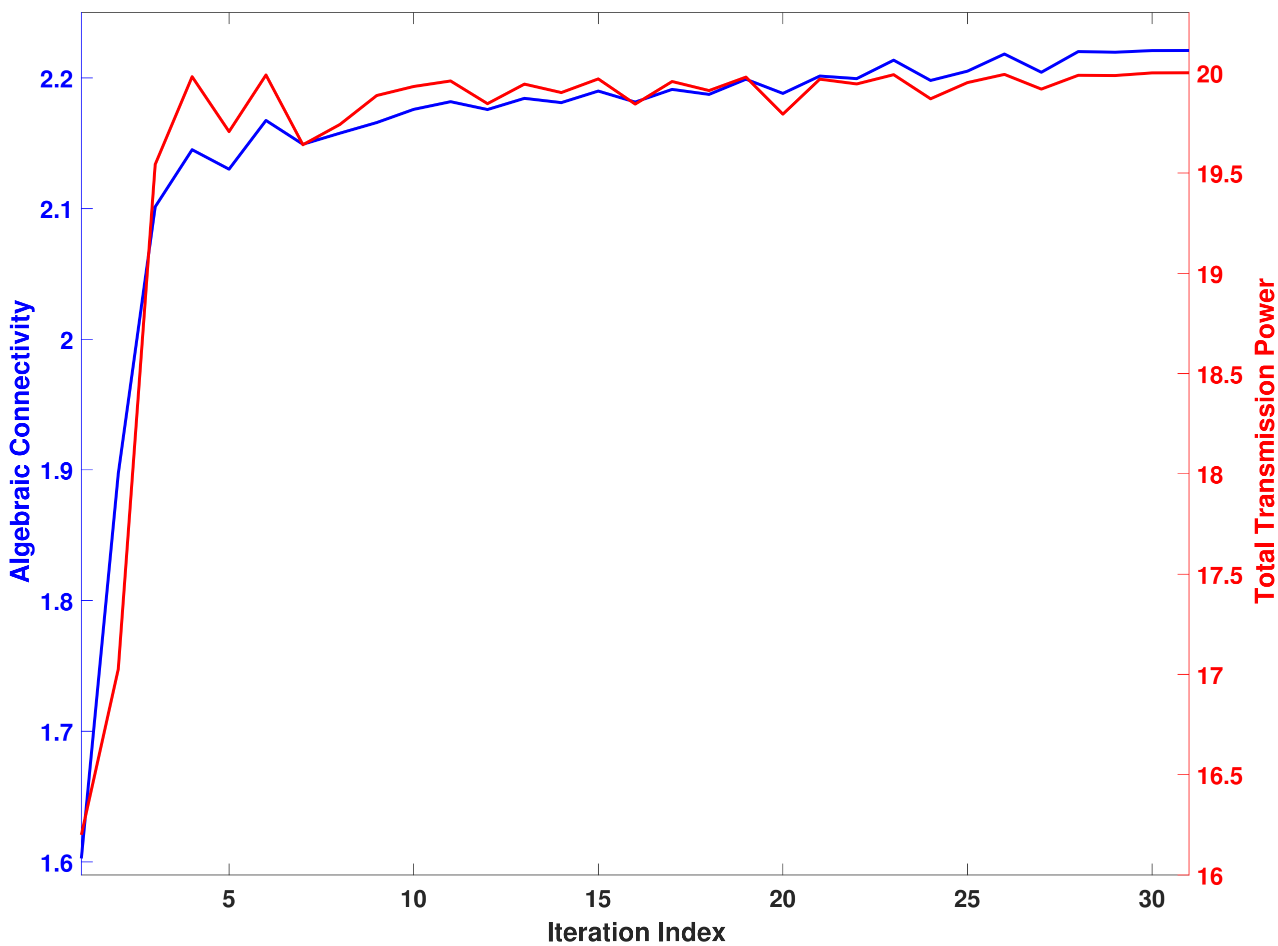

Figure 4.2: Evolution of the GAC and the total transmission power of the network of Example 4.1 for the first optimization problem in (4.7) ($l = 1$) obtained using the centralized algorithm.

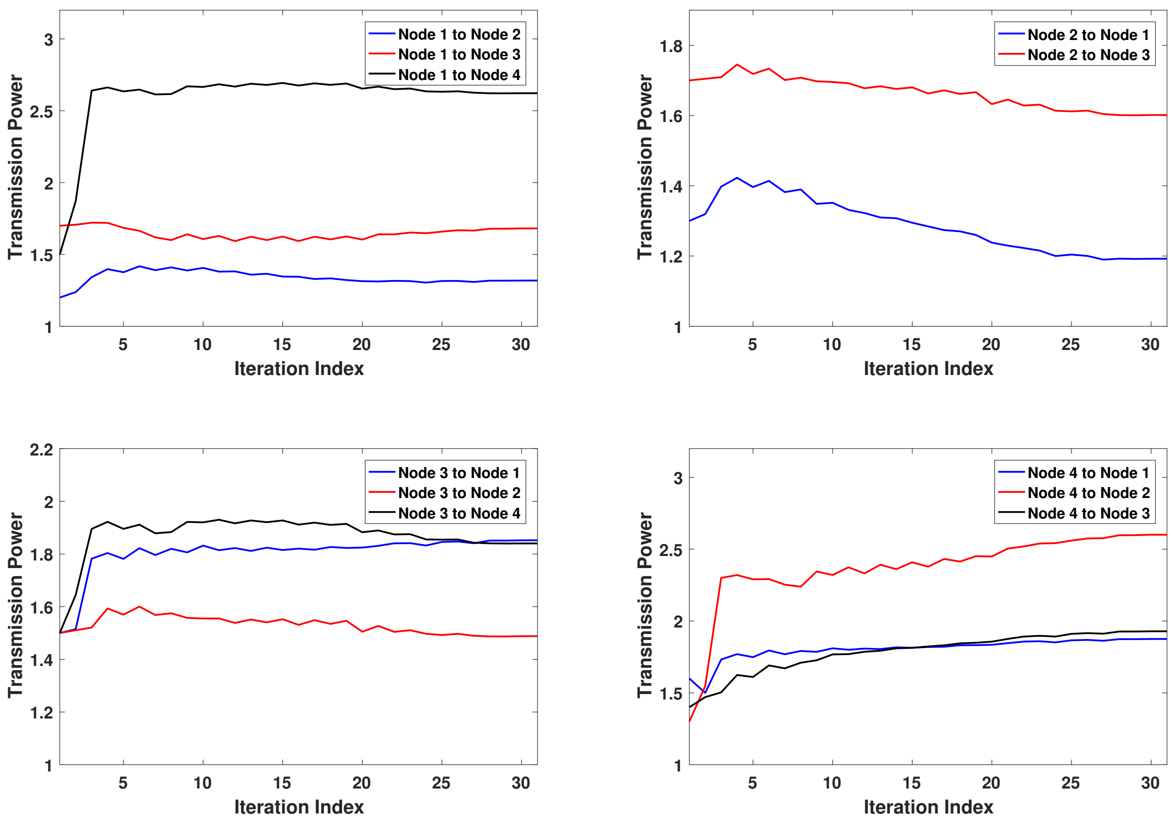

Figure 4.3: Evolution of the transmission power for the nodes comprising the network of Example 4.1 for the first optimization problem in (4.7) ($l = 1$) obtained using the centralized algorithm.

after a few iterations. Nonetheless, the following design parameters are chosen to find the best possible solution using Lemma 4.3: $m_{max} = 5$, $\nu = 0.005, \theta = 0.97, \alpha_{max} = 1$, $\beta = 0.2$, $\mu_0 = 5$, $\Gamma^0 = 0.1$, $\epsilon = \zeta = 10^{-4}$, and $\mathbf{v}(k) = -\mathbf{g}^{-}(k)$ for all $k \in \mathbb{N}$. The following optimal transmission power matrix is obtained

$$\mathbf{P}^*_{d,1} = \begin{bmatrix} 0 & 1.7249 & 1.7225 & 1.7101 \\ 1.7311 & 0 & 1.7272 & 1.7471 \\ 1.7245 & 1.7250 & 0 & 1.7323 \\ 1.7680 & 0 & 1.7392 & 0 \end{bmatrix},$$

resulting in the weight matrix

$$\mathbf{W}^*_{d,1} = \begin{bmatrix} 0 & 0.4923 & 0.6919 & 0.5288 \\ 0.5908 & 0 & 0.6527 & 0.5438 \\ 0.5295 & 0.5754 & 0 & 0.5884 \\ 0.7048 & 0 & 0.5518 & 0 \end{bmatrix}.$$

The corresponding network GAC is $\tilde{\lambda}(\mathbf{P}^*_{d,1}) = 1.8116$, and the total transmission power is $\sum_{i \in \mathbb{N}_n} \sum_{j \in \mathbb{N}_n} {P_{ij}}^*_{d,1} = 19.0579$. It is evident from the results of the distributed solution that it is terminated prematurely compared to the centralized algorithm. Additionally, it can be seen from the matrix $\mathbf{P}^*_{d,1}$ that all of the nodes are converging to the same transmission power, which is due to the initial weight Γ^0 given to the penalty function $I_l(\mathbf{P}(k))$, for $l = 1$. The parameter Γ^0 gives more weight to the penalty terms initially, and the optimization algorithm prioritizes minimizing the sum of these terms rather than giving more weight to the maximization of the GAC. If a bigger value is considered for Γ^0, even fewer iterations are completed successfully by the algorithm due to Lemma 4.3 not being satisfied. For this specific example, Lemma 3 is not applicable, but it can still be useful for other examples. If the method of [33] was utilized to obtain the GAC to be used with the exact backtracking line search (which can still be implemented in a distributed manner but will be computationally heavier), the obtained results via the distributed approach would be identical to its centralized counterpart, considering the same design parameters as in the centralized algorithm.

The *fmincon* function of MATLAB® is utilized to solve the optimization problem (4.7) for $l = 1$, with the same initial conditions, and appropriate penalty functions are considered for the violation of the constraints. The output of this function is

$$\mathbf{P}^*_{f,1} = \begin{bmatrix} 0 & 1.0012 & 1.7608 & 1.9681 \\ 1.0000 & 0 & 1.4689 & 2.6781 \\ 1.5712 & 1.8078 & 0 & 2.0850 \\ 2.8624 & 0 & 1.7965 & 0 \end{bmatrix},$$

which leads to a network GAC of $\tilde{\lambda}(\mathbf{P}^*_{f,1}) = 2.2211$ and a total transmission power of $\sum_{i\in\mathbb{N}_n}\sum_{j\in\mathbb{N}_n} P_{ij}{}^*_{f,1} = 20$.

It is evident from the above results that as a characteristic of the interior point method, neither the centralized algorithm nor the distributed algorithm violate any of the constraints. Comparing the results of the centralized algorithm and the *fmincon* function, it is observed that the minimum obtained by both methods are almost identical. The error between the minimum point obtained via the centralized algorithm and the MATLAB® function is less than 1%, which is within the numerical error range. The difference in the elements of the optimal transmission power matrix obtained via the two approaches may also be avoided by further tuning of the design parameters of the centralized algorithm. Note that using the exact GAC for determining the step size of the distributed algorithm results in the exact same solution as that of the centralized algorithm. Even though a higher GAC may require higher transmission power, it can be observed from Fig. 4.3 and $\mathbf{P}^*_{f,1}$ that the transmission powers of certain links have actually decreased. Similar to [28] and [29], this can be attributed to the asymmetric nature of the network and how the characteristics of those specific communication links are different from other links.

4.5.2 Minimizing $\tilde{f}_2$ in (4.7)

To implement the centralized optimization algorithm for $l = 2$ in the optimization problem (4.7), i.e., minimizing the total transmission power with constraints on the GAC and transmission power levels, the following design parameters are considered: $m_{max} = 25$, $\nu = 0.05, \theta = 0.7, \alpha_{max} = 1$, $\beta = 0.1$, $\mu_0 = 2$, $\Gamma^0 = 85$, $\epsilon = \zeta = 10^{-4}$, and $\mathbf{v}(k) = -\mathbf{g}^{-}(k)$ for all $k \in \mathbb{N}$. The following optimal transmission power matrix is obtained as

$$\mathbf{P}^*_{c,2} = \begin{bmatrix} 0 & 1.0007 & 1.2477 & 1.0012 \\ 1.0011 & 0 & 1.0635 & 1.0007 \\ 1.2847 & 1.2199 & 0 & 1.0043 \\ 1.4747 & 0 & 1.0005 & 0 \end{bmatrix},$$

resulting in the weight matrix

$$\mathbf{W}^*_{c,2} = \begin{bmatrix} 0 & 0.4063 & 0.4428 & 0.2061 \\ 0.4816 & 0 & 0.4419 & 0.3968 \\ 0.3644 & 0.4040 & 0 & 0.2893 \\ 0.6099 & 0 & 0.4418 & 0 \end{bmatrix}.$$

The optimal total transmission power from the centralized algorithm is $\sum_{i\in\mathbb{N}_n}\sum_{j\in\mathbb{N}_n} P_{ij}{}^*_{c,2} = 12.2990$, and the corresponding network GAC is $\tilde{\lambda}(\mathbf{P}^*_{c,2}) = 1.5$. The evolution of the total transmission power and the GAC of the network is presented in Fig. 4.4. In addition, Fig. 4.5 depicts the evolution of the transmission power levels of each node.

To implement the distributed optimization Algorithm 4.3 for $l = 2$ in the optimization problem (4.7), the following design parameters are considered: $m_{max} = 25$, $\nu = 0.01, \theta = 0.7, \alpha_{max} = 1$, $\beta = 0.1$, $\mu_0 = 4$, $\Gamma^0 = 85$, $\epsilon = \zeta = 10^{-4}$, $L = 1.5$ and $\mathbf{v}(k) = -\mathbf{g}^{-}(k)$ for all $k \in \mathbb{N}$. Unlike the previous case (the first optimization problem in Example 4.1), the problem with the approximate backtracking algorithm pointed out

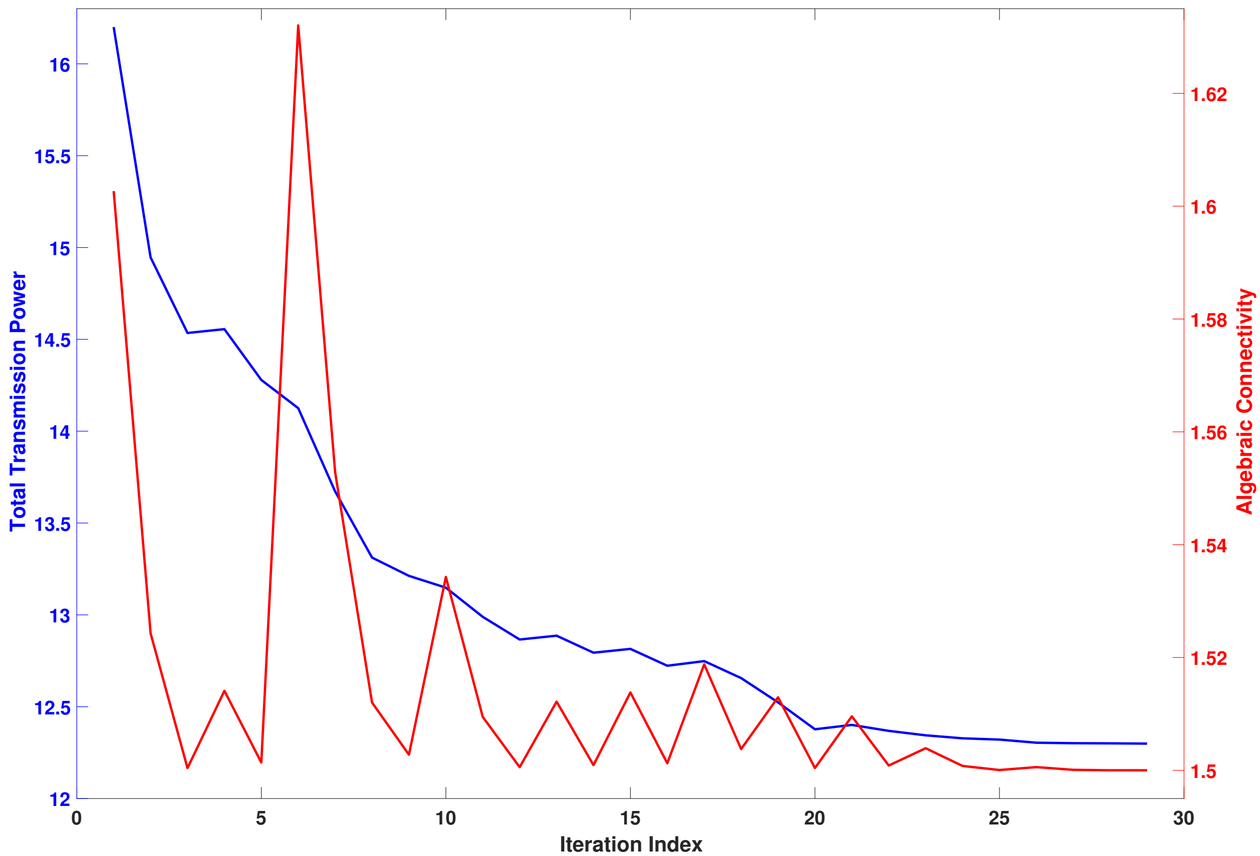

Figure 4.4: Evolution of the total transmission power and the GAC of the network of the network of Example 4.1 for the second optimization problem in (4.7) ($l = 2$) obtained using the centralized algorithm.

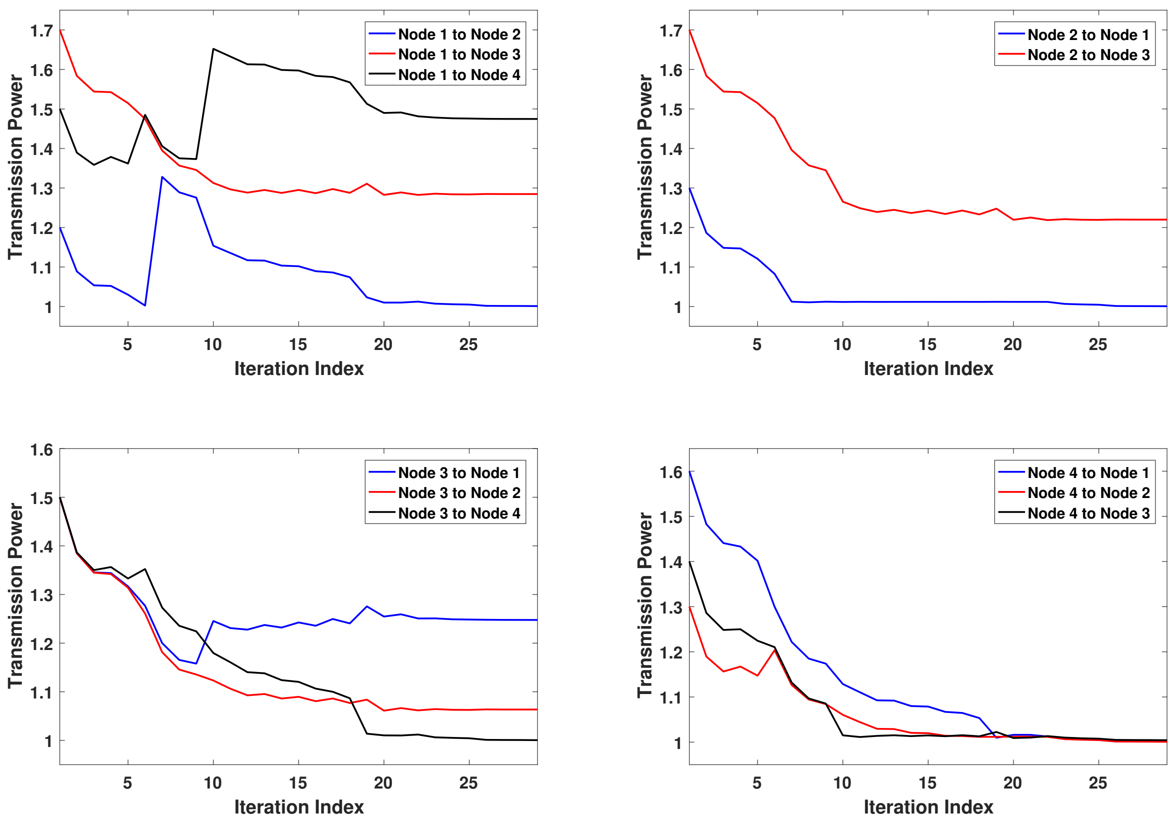

Figure 4.5: Evolution of the transmission power for the nodes comprising the network of Example 4.1 for the second optimization problem in (4.7) ($l = 2$) obtained using the centralized optimization algorithm.

in Remark 4.4 does not occur in using Lemma 4.4 in this case. The following optimal transmission power matrix is obtained

$$\mathbf{P}^*_{d,2} = \begin{bmatrix} 0 & 1.0018 & 1.2433 & 1.0541 \\ 1.0015 & 0 & 1.0037 & 1.0022 \\ 1.3686 & 1.2168 & 0 & 1.0021 \\ 1.4053 & 0 & 1.0062 & 0 \end{bmatrix},$$

resulting in the weight matrix

$$\mathbf{W}^*_{d,2} = \begin{bmatrix} 0 & 0.4065 & 0.4401 & 0.2314 \\ 0.4817 & 0 & 0.4166 & 0.3972 \\ 0.3989 & 0.4028 & 0 & 0.2882 \\ 0.5832 & 0 & 0.4430 & 0 \end{bmatrix}.$$

The optimal total transmission power obtained by the distributed algorithm is $\sum_{i \in \mathbb{N}_n} \sum_{j \in \mathbb{N}_n} P_{ij}{}^*_{d,2} = 12.3056$, and the corresponding network GAC is $\tilde{\lambda}(\mathbf{P}^*_{d,2}) = 1.5000$. The evolution of the total transmission power and the GAC of the network using the proposed distributed algorithm is provided in Fig. 4.6. The evolution of the various transmission powers of each node is presented in Fig. 4.7.

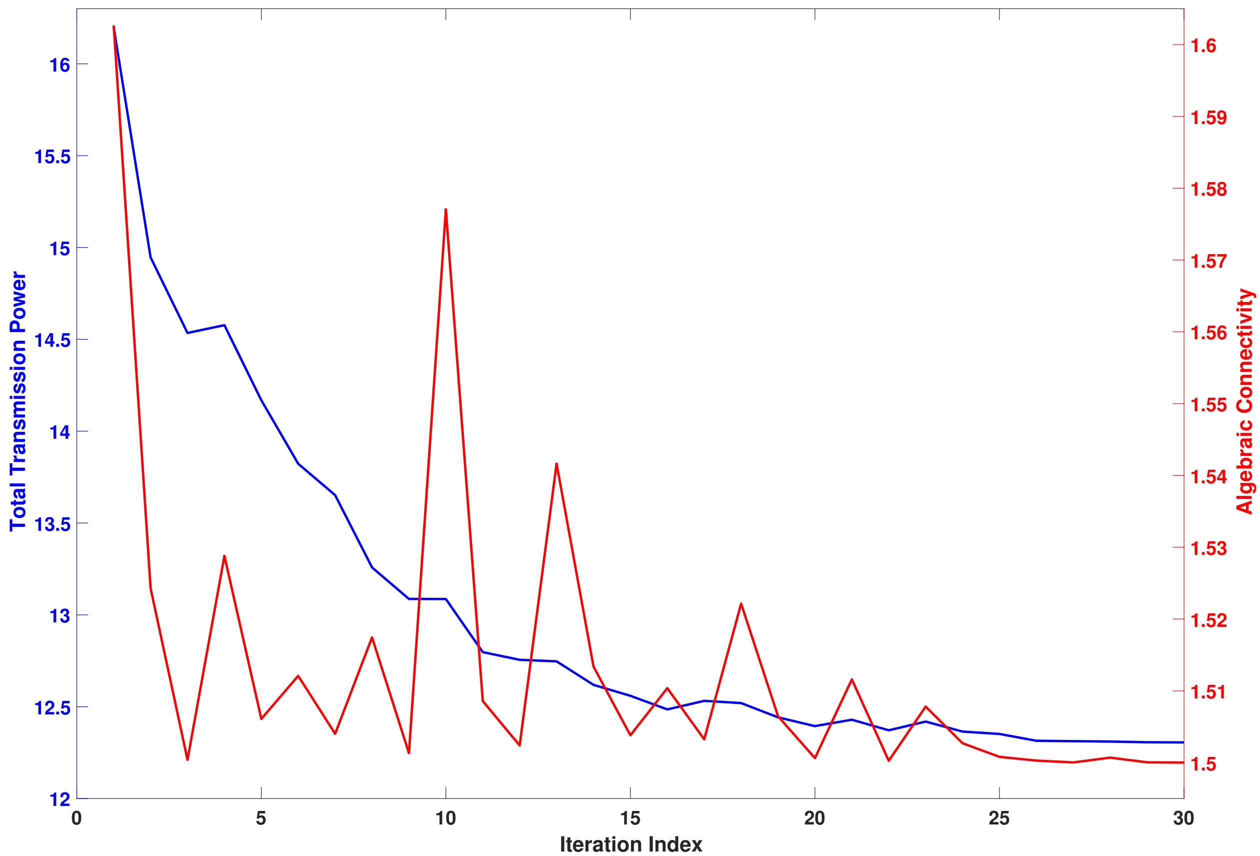

Figure 4.6: Evolution of the total transmission power and the GAC of the network of the network of Example 4.1 for the second optimization problem in (4.7) ($l = 2$) obtained using the distributed algorithm.

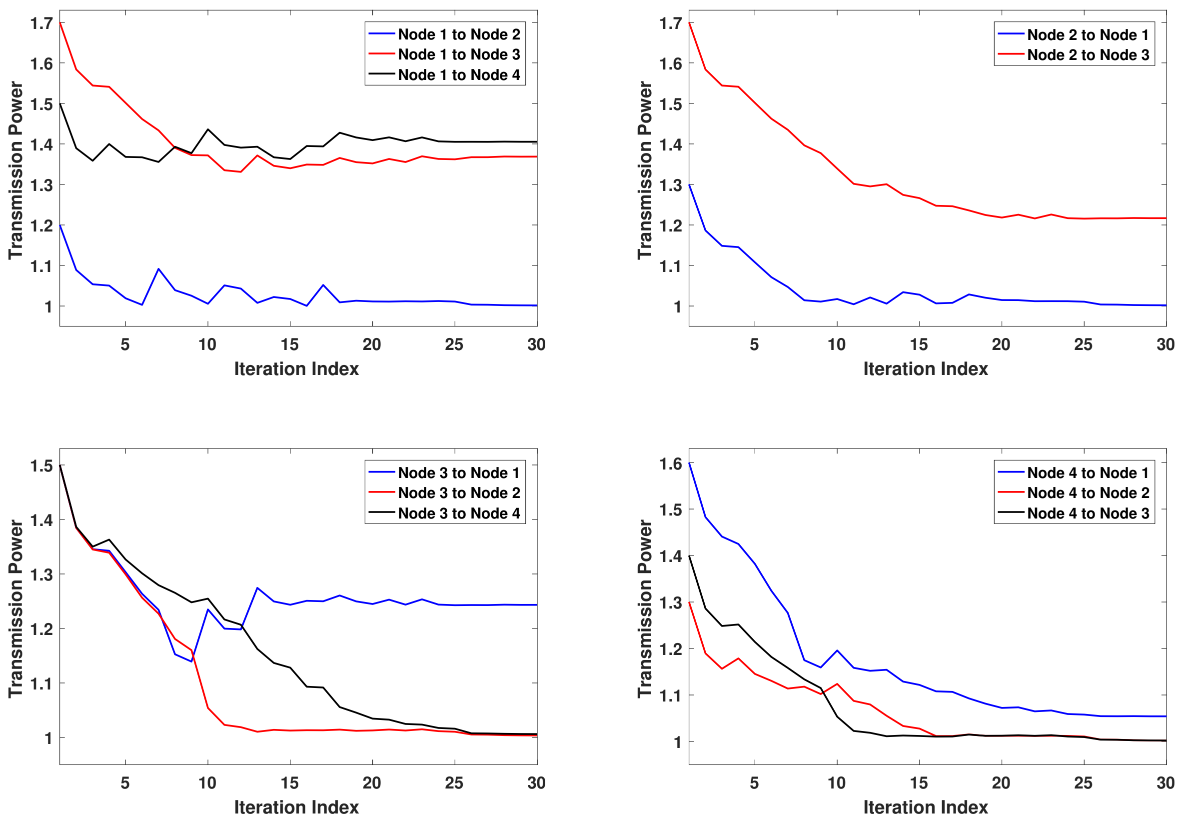

Figure 4.7: Evolution of the transmission power for the nodes comprising the network of Example 4.1 for the second optimization problem in (4.7) ($l = 2$) obtained using the distributed algorithm.

The output of the *fmincon* function for $l = 2$, given the same initial values is

$$\mathbf{P}^*_{f,2} = \begin{bmatrix} 0 & 1.0000 & 1.3837 & 1.0000 \\ 1.0003 & 0 & 1.1572 & 1.0000 \\ 1.2935 & 1.0278 & 0 & 1.0022 \\ 2.4187 & 0 & 1.0000 & 0 \end{bmatrix},$$

which results in a total transmission power of $\sum_{i\in\mathbb{N}_n}\sum_{j\in\mathbb{N}_n} P_{ij}{}^*_{f,2} = 12.2833$ and a network GAC of $\tilde{\lambda}(\mathbf{P}^*_{f,2}) = 1.5001$.

The discussions given at the end of the previous subsection (for $l = 1$) hold true for this case as well. As observed from Figs. 4.4-4.7, the proposed algorithms do not violate any of the constraints. Similar to the previous subsection, the difference between the optimal points obtained via the different approaches is within 1%. Note that the elements of the matrices $\mathbf{P}^*_{c,2}$ and $\mathbf{P}^*_{d,2}$ are close to each other, and with a proper choice of design parameters, they could be even closer. Ultimately, for the considered optimization problem where smaller transmission powers are desirable, it can be seen from $\mathbf{P}^*_{f,2}$ that this objective is achieved by increasing the power corresponding to the communication link $\vec{14} \in E$.

4.5.3 Minimizing $\tilde{f}_3$ in (4.7)

To implement the centralized optimization algorithm for $l = 3$ in the optimization problem (4.7), i.e., maximizing the network lifetime with constraints on the GAC and transmission power levels, the following design parameters are considered: $m_{max} = 40$, $\nu = 0.001, \theta = 0.95, \alpha_{max} = 1$, $\beta = 0.05$, $\mu_0 = 1.3$, $\Gamma^0 = 20$, $\epsilon = \zeta = 10^{-4}$, and $\mathbf{v}(k) = -\mathbf{g}^{-}(k)$ for all $k \in \mathbb{N}$. The following optimal transmission power matrix is obtained

$$\mathbf{P}^*_{c,3} = \begin{bmatrix} 0 & 1.4678 & 1.4077 & 1.0122 \\ 1.0015 & 0 & 1.4731 & 1.2136 \\ 1.0002 & 1.7721 & 0 & 1.3196 \\ 1.1685 & 0 & 1.2446 & 0 \end{bmatrix},$$

resulting in the weight matrix

$$\mathbf{W}^*_{c,3} = \begin{bmatrix} 0 & 0.4666 & 0.5389 & 0.2114 \\ 0.4817 & 0 & 0.5854 & 0.4471 \\ 0.2407 & 0.5885 & 0 & 0.4343 \\ 0.4787 & 0 & 0.4852 & 0 \end{bmatrix}.$$

The optimal lifetimes obtained by the centralized algorithm are

$$T(\mathbf{P}^*_{c,3}) = [15.3793, 15.9550, 15.3807, 15.3813]^{\mathrm{T}},$$

and the corresponding network GAC is $\tilde{\lambda}(\mathbf{P}^*_{c,3}) = 1.5032$. The values of the lifetime and the GAC of the network as the iteration index k in the proposed centralized algorithm increases are shown in Fig. 4.8. The evolution of the lifetime of the individual nodes is presented in Fig. 4.9, and the way the different transmission power levels of each node changes is depicted in Fig. 4.10.

To implement the distributed optimization algorithm (Algorithm 4.3) for $l = 3$ in the optimization problem (4.7), the following design parameters are considered: $m_{max} = 30$, $\nu = 0.001, \theta = 0.9, \alpha_{max} = 1$, $\beta = 0.1$, $\mu_0 = 1.5$, $\Gamma^0 = 10$, $\epsilon = \zeta = 10^{-4}$, $L = 1.5$ and $\mathbf{v}(k) = -\mathbf{g}^-(k)$ for all $k \in \mathbb{N}$. Similar to the previous case ($l = 2$), the problem encountered while using the approximate backtracking algorithm in subsection 4.5.1 (concerning Lemma 4.3) does not arise in this case ($l = 3$) using Lemma 4.4. The resulting optimal transmission power matrix is

$$\mathbf{P}^*_{d,3} = \begin{bmatrix} 0 & 1.6176 & 1.3686 & 1.0077 \\ 1.0002 & 0 & 1.4831 & 1.2027 \\ 1.0003 & 1.8292 & 0 & 1.3650 \\ 1.1596 & 0 & 1.2605 & 0 \end{bmatrix},$$

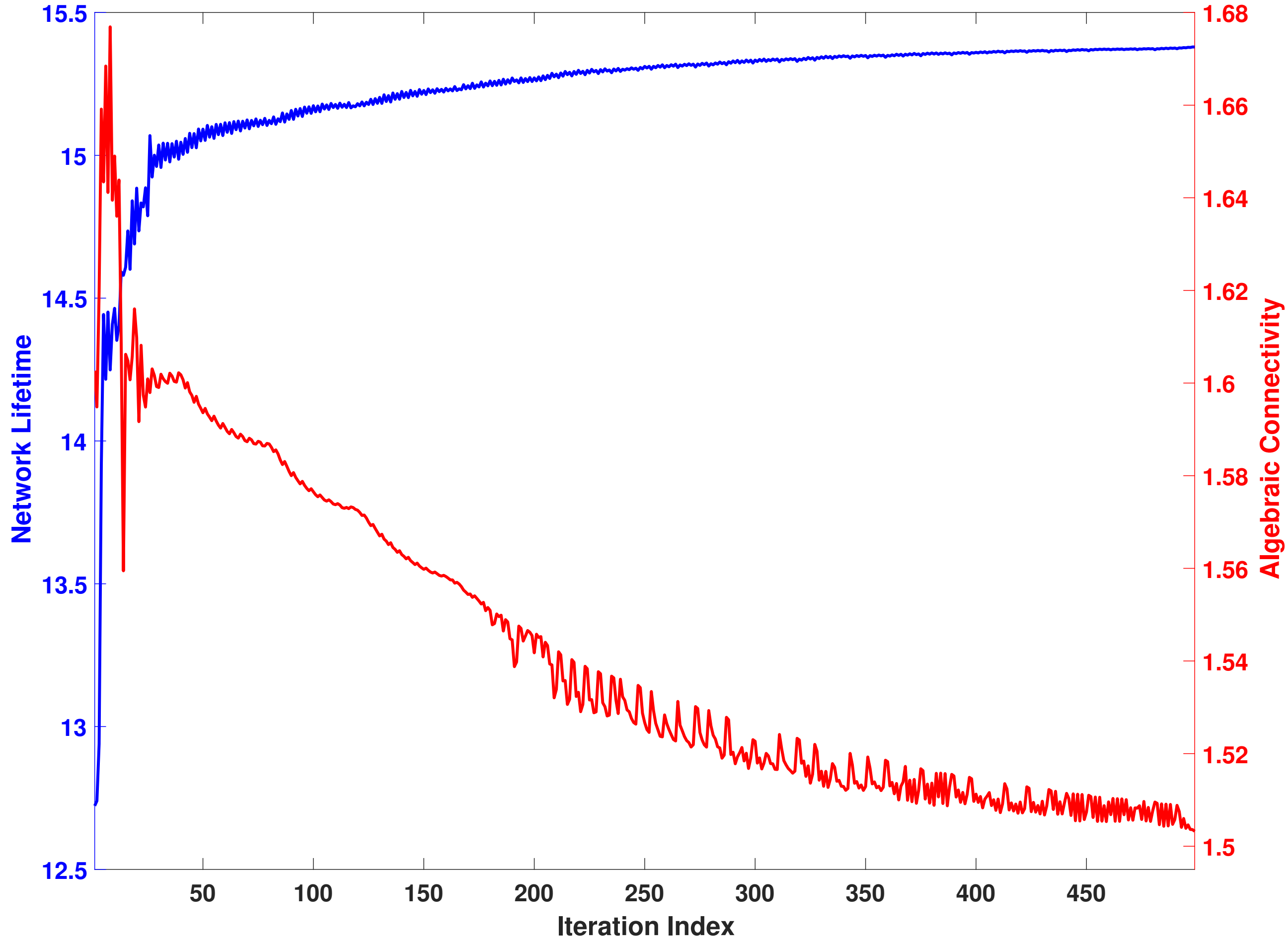

Figure 4.8: Evolution of the lifetime and the GAC of the network of the network of Example 4.1 for the third optimization problem in (4.7) ($l = 3$) obtained using the centralized algorithm.

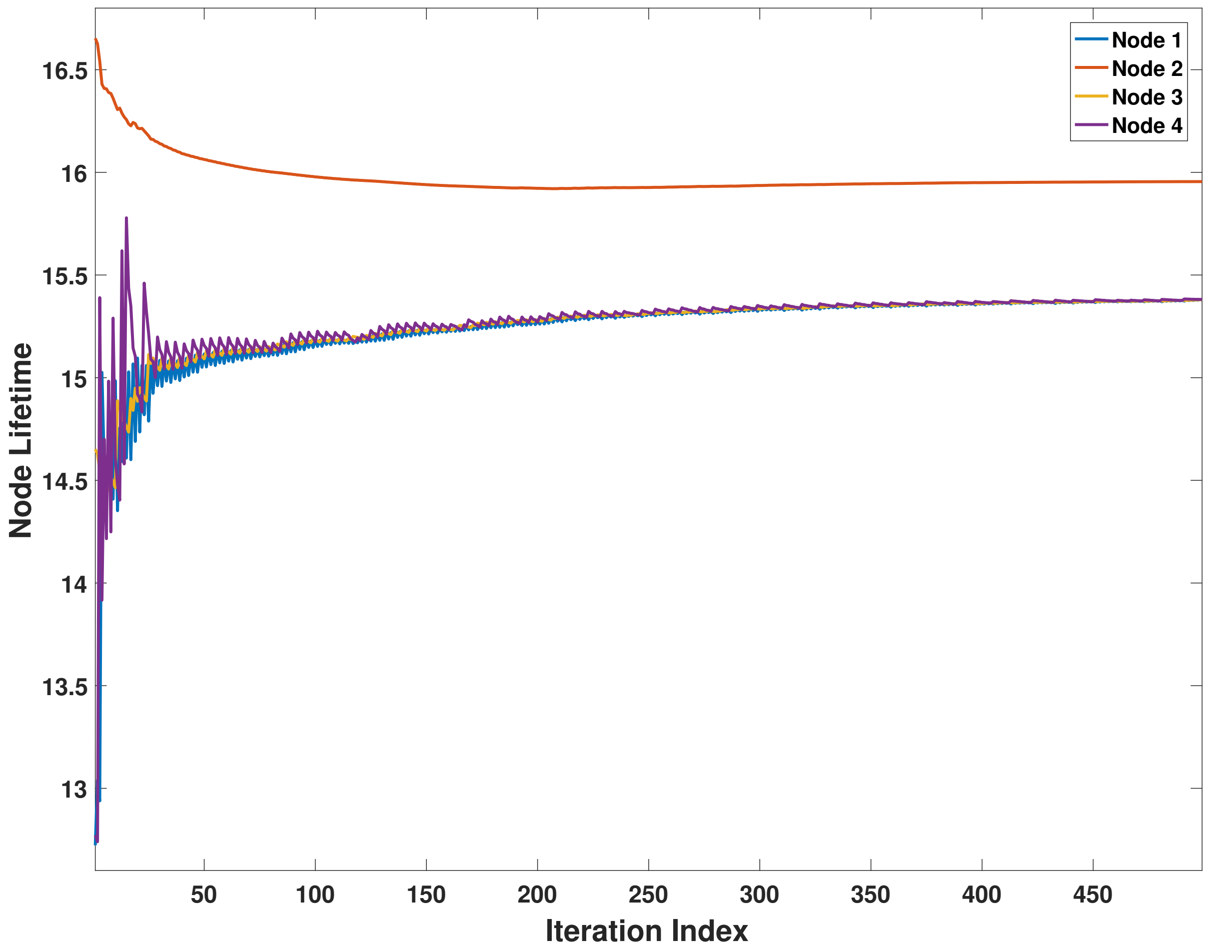

Figure 4.9: Evolution of the lifetime of the nodes of the network of the network of Example 4.1 for the third optimization problem in (4.7) ($l = 3$) obtained using the centralized algorithm.

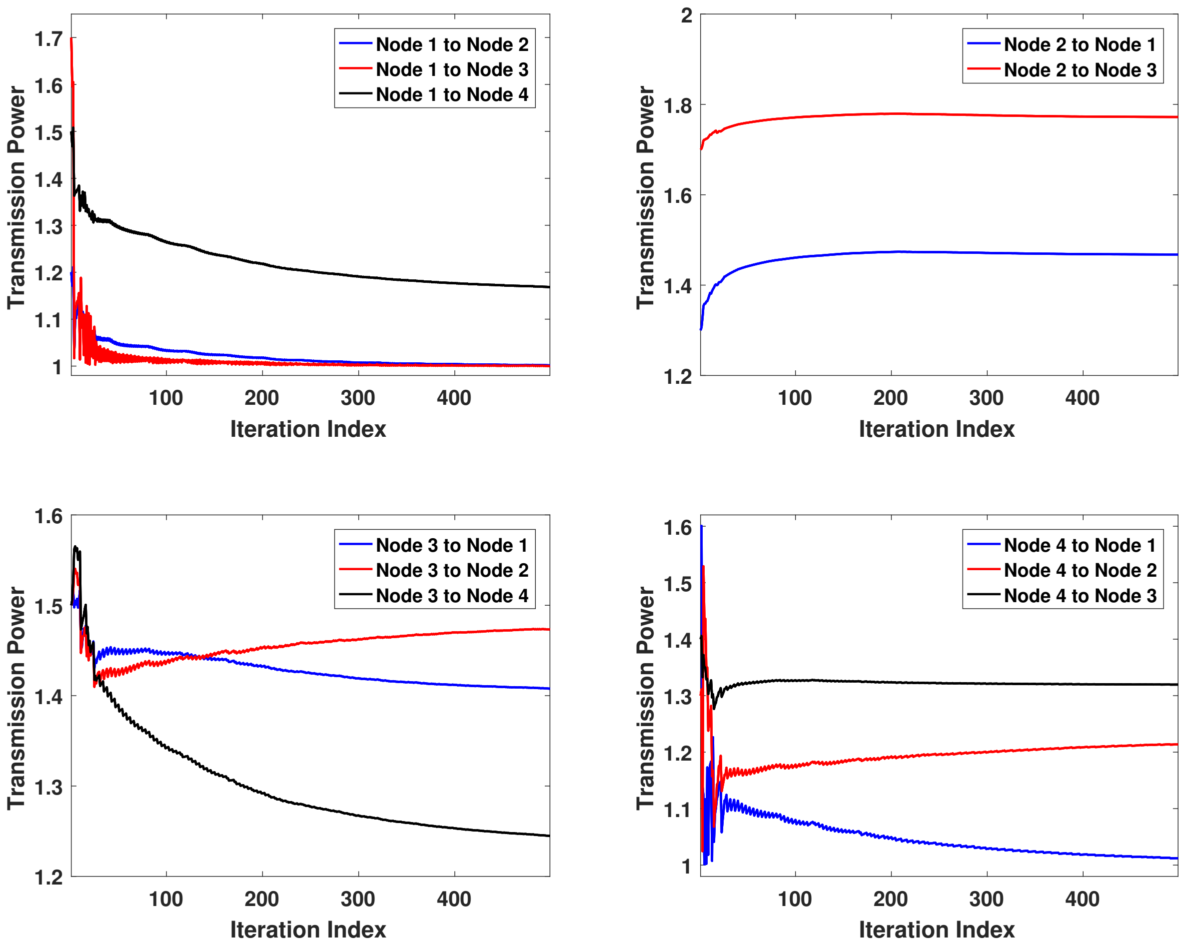

Figure 4.10: Evolution of the transmission power for the nodes comprising the network of Example 4.1 for the third optimization problem in (4.7) ($l = 3$) obtained using the centralized algorithm.

leading to the weight matrix

$$\mathbf{W}_{d,3}^{*} = \begin{bmatrix} 0 & 0.4821 & 0.5165 & 0.2092 \\ 0.4815 & 0 & 0.5884 & 0.4447 \\ 0.2407 & 0.6039 & 0 & 0.4534 \\ 0.4743 & 0 & 0.4877 & 0 \end{bmatrix}.$$

The optimal lifetimes obtained by the distributed algorithm are

$$T(\mathbf{P}_{d,3}^{*}) = [15.3934, 15.3942, 15.3950, 15.3962]^{\mathrm{T}},$$

and the corresponding network GAC is $\tilde{\lambda}(\mathbf{P}_{d,3}^{*}) = 1.5013$. Fig. 4.11 shows the evolution of the lifetime and the GAC of the network as the iteration index k in the distributed optimization algorithm increases. The evolution of the lifetime of the individual nodes is presented in Fig. 4.12, and the way the different transmission power levels of each node changes is depicted in Fig. 4.13.

The output of the *fmincon* function in this case with the same initial variables and parameters as before, is

$$\mathbf{P}_{f,3}^{*} = \begin{bmatrix} 0 & 1.9915 & 1.4232 & 1.0000 \\ 1.0000 & 0 & 1.6723 & 1.0000 \\ 1.0000 & 1.2699 & 0 & 2.0552 \\ 1.1248 & 0 & 1.0001 & 0 \end{bmatrix},$$

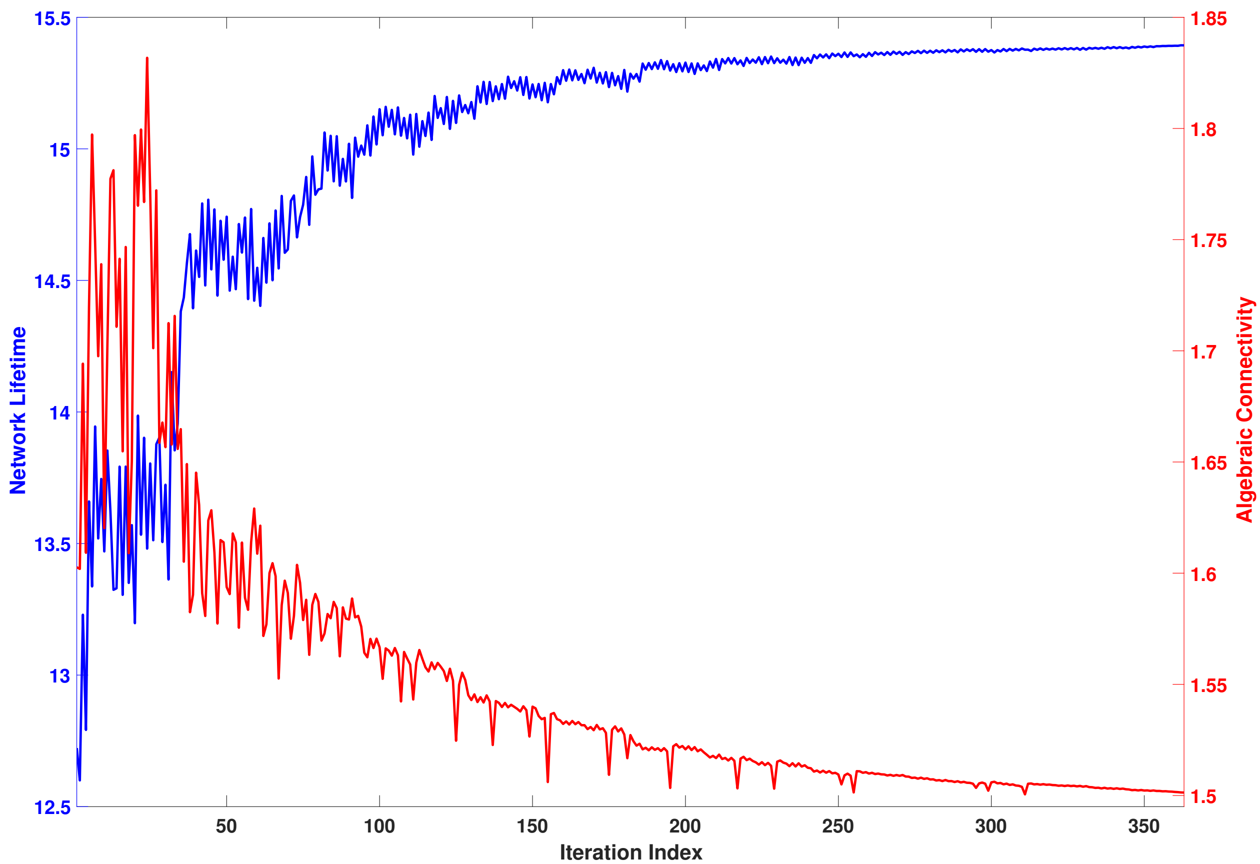

Figure 4.11: Evolution of the lifetime and the GAC of the network of the network of Example 4.1 for the third optimization problem in (4.7) ($l = 3$) obtained using the distributed algorithm.

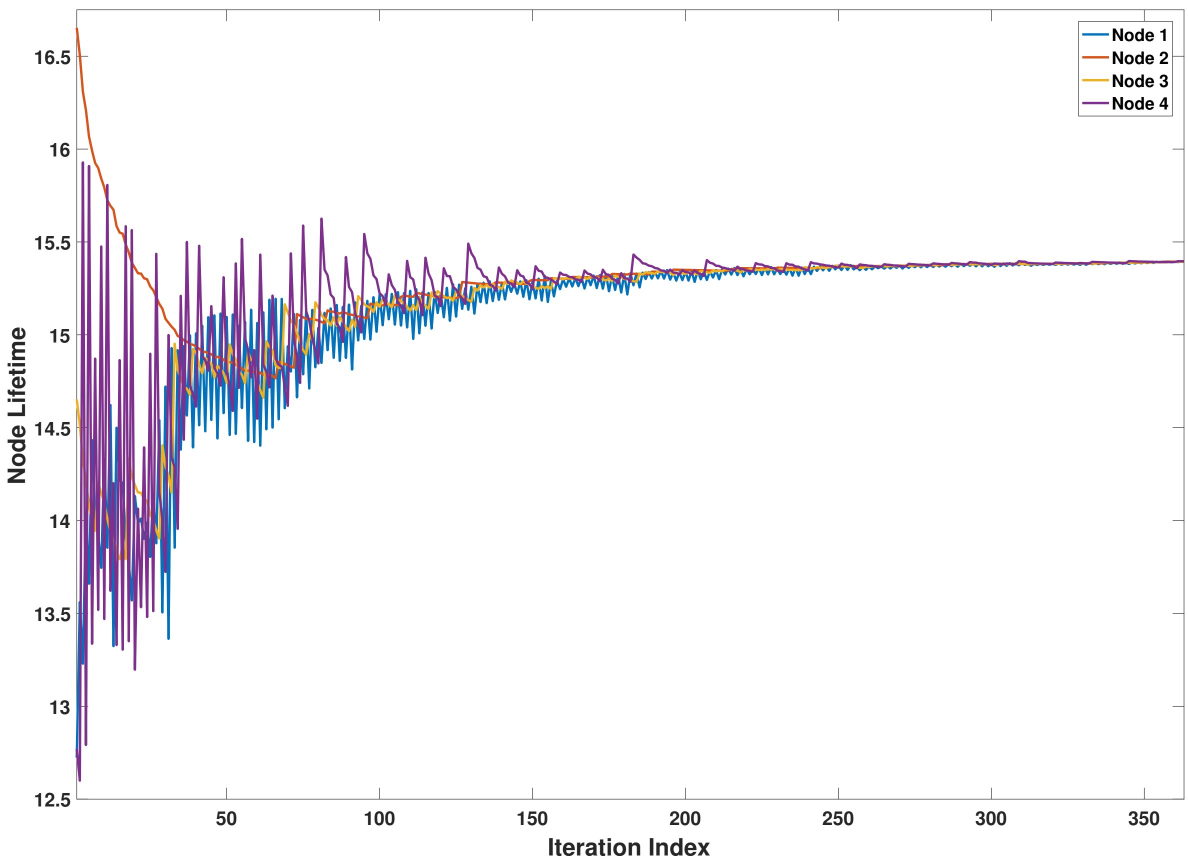

Figure 4.12: Evolution of the lifetime of the nodes of the network of the network of Example 4.1 for the third optimization problem in (4.7) ($l = 3$) obtained using the distributed algorithm.

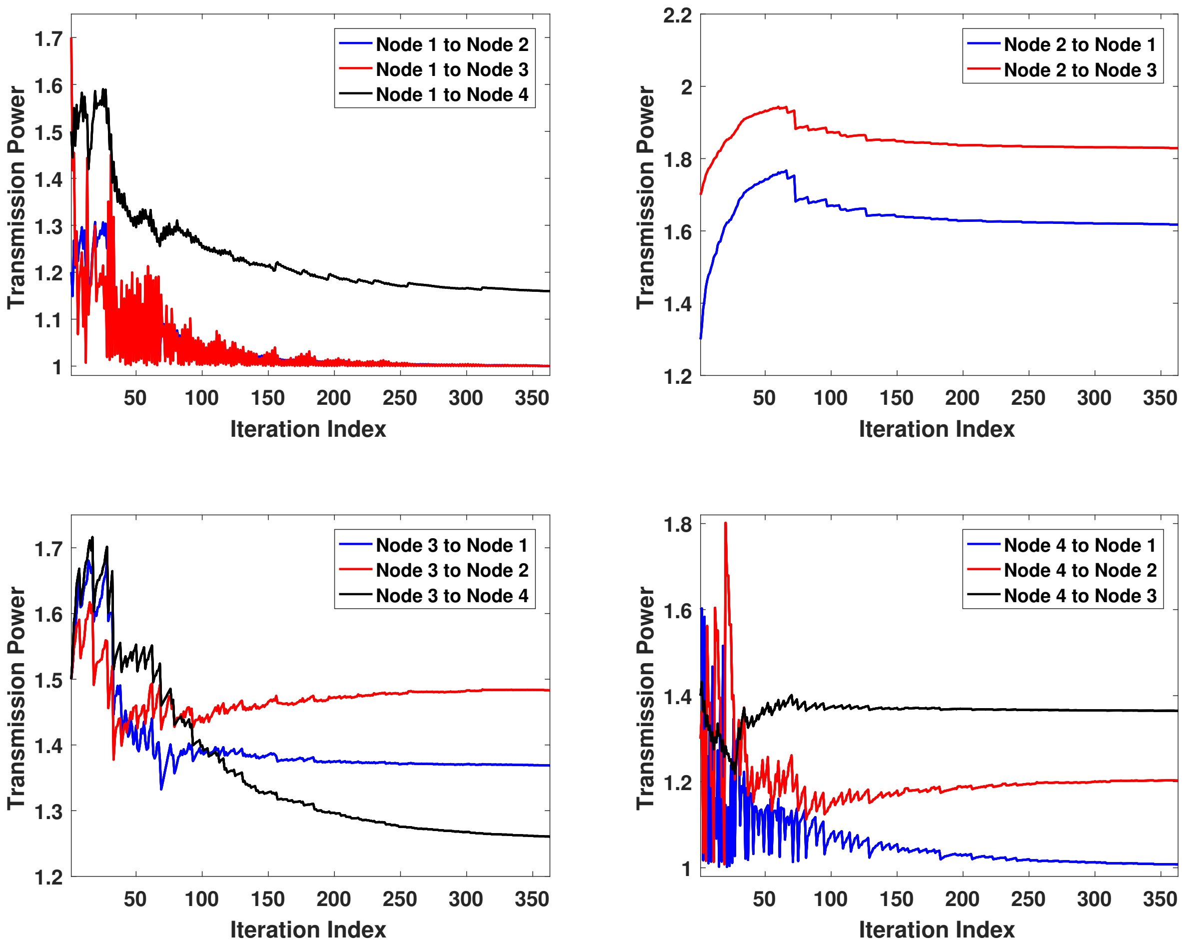

Figure 4.13: Evolution of the transmission power for the nodes comprising the network of Example 4.1 for the third optimization problem in (4.7) ($l = 3$) obtained using the distributed algorithm.

and the resultant network GAC is $\tilde{\lambda}(\mathbf{P}^*_{f,3}) = 1.5000$. The obtained optimal lifetimes are

$$T(\mathbf{P}^*_{f,3}) = [15.4443, 15.4579, 15.4498, 15.4509]^{\mathrm{T}}.$$

The discussions presented in the previous two subsections to justify the simulation results for $l = 1, 2$ hold true for $l = 3$ as well. That is, according to Figs. 4.8, 4.10, and Figs. 4.11, 4.13, none of the constraints is violated. The discrepancy between the above values and those obtained by using the proposed algorithm is less than 1%, and is due to the numerical inaccuracies. It can also be observed that to maximize the network lifetime, the transmission powers associated with certain communication links are actually increased.

Note that the transmission power matrices $\mathbf{P}^*_{c,2}$, $\mathbf{P}^*_{d,2}$ and $\mathbf{P}^*_{f,2}$ corresponding to the optimization problem P2 result in the network lifetimes of 14.0598, 13.8949 and 14.1011, respectively, which are, as expected, shorter than the solution of the optimization problem P3.

In terms of convergence time, the run-times of the centralized algorithm and the *fmincon* function for all the considered optimization problems are almost identical to each other, and both take just a few seconds. On the other hand, the distributed algorithm takes a few minutes to converge to the same optimum. As previously mentioned, the most computationally heavy element of the distributed algorithm is the estimation of the GAC using only local information. Apart from that, the run-time of the other parts of the distributed algorithm would be similar to the analogous parts in the other two approaches.

4.6 Conclusion

In this study, three optimization problems are considered over an asymmetric network, represented by a weighted directed graph. In the first problem, it is desired to maximize the generalized algebraic connectivity (GAC) while satisfying certain constraints on the transmission power of the network and that of each node. The second problem is to minimize the total transmission power of the network while imposing a constraint on the GAC of the network as well as constraints on the transmission power levels. The third one is the problem of network lifetime maximization subject to constraints on the GAC and the power levels is considered. Due to the complexity of the aforementioned problems, they are solved numerically via the interior point method, transforming the constrained optimization problems into sequential unconstrained problems. The subgradient method with backtracking line search is adopted to solve each interior point subproblem. To implement this approach in a distributed manner, certain modifications are made to the centralized algorithm, e.g., a finite-time algorithm is used to obtain the exact GAC in a distributed manner, and the backtracking algorithm is approximated in order not to require the GAC estimation in determining the step size. It is proved analytically that the proposed distributed algorithm converges to a local or global optimum.

Efficacy of the proposed method is verified by numerical simulations. It is observed that the errors between the optimal solutions obtained via the proposed algorithm and the *fmincon* function of MATLAB® are all less than 1%. As a counter-intuitive result, it was observed that in order to increase the network GAC, the transmission powers associated with certain communication links had to be decreased. Additionally, it was observed that in order to increase the network lifetime, certain transmission powers had to be increased.

In this work, it was assumed that the network is structurally static, and that the GAC is observable to nodes. As a future work, one can incorporate the requirements for observability of the GAC in the optimization problems. This enables one to add or remove communication links while preserving the GAC observability from each node. Additionally, considering a dynamic nonlinear behavior for the battery depletion of the nodes would be another contribution to the current study. Finally, a more general approximate backtracking algorithm, suitable for all applications, could be developed.

Chapter 5

Conclusions and Future Work

In this thesis, three different problems concerning proper power management are investigated over asymmetric networks represented by weighted directed graphs. In all the problems, the notion of the generalized algebraic connectivity (GAC) is used as the network connectivity measure and is formulated as an implicit function of the nodes' transmission powers. Lifetime of the network is also formulated as a function of nodes' transmission powers and is defined as the minimum lifetime over all nodes. It is assumed that the nodes deplete their battery linearly with respect to the transmission powers used for communication with their neighbors.

In Chapter 2, it is desired to minimize the total transmission power of the network while having constraints on the minimum acceptable connectivity level for the network and the individual transmission powers. In this chapter, it is assumed that each node uses the same transmission power to communicate with its out-neighbor set. The interior point method is used to transform the inequality-constrained optimization problem into a sequential unconstrained problem. To solve each subproblem, the subgradient method is used to obtain the search directions at each optimization iteration. Since the subgradient method may not necessarily produce a descent direction, this issue is addressed. The backtracking line search is then used to obtain the step-sizes to move along the search directions. Asymptotic convergence of the proposed algorithm is then demonstrated analytically. An experimental underwater acoustic sensor network (UWASN) is considered as an asymmetric network to verify the effectiveness of the proposed algorithm. The results of the proposed algorithm are compared to the output of the *fmincon* function of MATLAB®, and it is seen that the maximum discrepancy between the elements of the optimal transmission power vectors obtained via the two methods is less than 1%, which is within the numerical error range.

In Chapter 3, the problem of network lifetime maximization with constraints on the values of the transmission powers and minimum acceptable network connectivity level is investigated. An additional constraint is added requiring the nodes to deplete all their energies simultaneously,i.e., all the nodes will have the same lifetime. Given this equality constraint, the mixed interior point-exterior point method is used to transform the constrained optimization problem into a sequential unconstrained problem. Similar to Chapter 2, the subgradient method with the backtracking line search is used again to solve the subproblems. Asymptotic convergence of the algorithm is shown analytically, and its effectiveness is verified by simulations where the same network from Chapter 2 is used again as an example of an asymmetric network. Since it is assumed in this chapter that the edge set of the network may change, it is observed that certain communication links have been removed from the initial set-up of the network. In general, it is seen that the transmission powers have decreased to increase the lifetime of the network; however, for certain communication links, the transmission powers have increased. This counter-intuitive result can be attributed to the asymmetric nature of the considered network. Given the obtained optimal transmission power matrix, lifetime of the network has increased almost 50% compared to the initial set-up.

In Chapter 4, the problem of network connectivity maximization with constraints on the total transmission power of the network and the individual node transmission powers is considered along with the problems of Chapter 2 and Chapter 3, and a distributed approach is presented to numerically solve them. Note that the requirement on nodes having the same lifetime in Chapter 3 is removed here to achieve a streamlined optimization algorithm using only the interior point algorithm. Using this method, the inequality-constrained optimization problems are transformed into sequential unconstrained problems. To solve the subproblems, the subgradient method is implemented in a distributed manner, and the estimation of global values such as the GAC values is taken into account. An approximate backtracking line search is proposed which does not require the estimation of new GAC values to obtain the step-sizes. Asymptotic convergence of the algorithm is demonstrated analytically, and its effectiveness is assessed by means of numerical simulations. The errors between the optimal solutions obtained via the proposed algorithm and those of the *fmincon* function of MATLAB® are all less than 1%. It is observed that the obtained results for some nodes are counter-intuitive, e.g., to increase the GAC of the network, the transmission powers corresponding to certain communication links have decreased.

5.1 Future Work

Considering that the distributed optimization algorithm presented in Chapter 4 is the most significant result of this thesis and contains the material of Chapters 2 and 3, the following are some suggestions to improve the results and to relax some of the assumptions of the distributed method for future work:

- To be able to estimate the global values of the GAC using a distributed algorithm such as [4] or [33], it is assumed that the eigenvalue corresponding to the GAC is observable to the nodes. Since it is desired to preserve this property as the optimization algorithm proceeds, it is also assumed that the network is structurally static, i.e., if the GAC is observable initially, it will be so for the rest of the optimization. For future work, one can incorporate the necessary observability conditions mentioned in [34,35] in the considered optimization problems. This will allow the topology of the network to change to a certain degree while preserving the GAC observability to the nodes.

- It is assumed in this thesis that the nodes deplete their batteries linearly with respect to the transmission powers used for communication. However, in reality, batteries have dynamic nonlinear behavior as discussed in [17]. The proposed distributed algorithm is not critically dependent on the considered battery model. Nevertheless, considering a nonlinear battery behavior would be another contribution to the present work.

- The approximate backtracking line search presented in Chapter 4 requires knowledge of the Lipschitz constant of the GAC function for the specific network topology. Additionally, as explained in Remark 4.4, it may not be applicable to every application. Even though the exact backtracking line search can be implemented in a distributed manner at the expense of higher computational time but without the issues of the approximate method, a more general approximate backtracking algorithm which does not require the calculation of new global GAC values could be developed for future work.

Bibliography

[1] M. M. Asadi, M. Khosravi, A. G. Aghdam, and S. Blouin, “Joint power optimization and connectivity control problem over underwater random sensor networks,” in *Proceedings of the 2013 Research in Adaptive and Convergent Systems*, 2013, pp. 277-282.

[2] M. M. Asadi, S. Blouin, and A. G. Aghdam, “Distributed connectivity optimization in asymmetric networks,” in *Proceedings of the 56th IEEE Conference on Decision and Control*, December 2017, pp. 82-87.

[3] P. Di Lorenzo, and S. Barbarossa, “Distributed estimation and control of algebraic connectivity over random graphs,” *IEEE Transaction on Signal Processing*, vol. 62, no. 21, pp. 5615-5628, 2014.

[4] M. M. Asadi, M. Khosravi, A. G. Aghdam, and S. Blouin, “A subspace consensus approach for distributed connectivity assessment of asymmetric networks,” in *Proceedings of the 55th IEEE Conference on Decision and Control*, December 2016, pp.

4302-4307.

[5] M. C. De Gennaro, and A. Jadbabaie,“Decentralized control of connectivity for multi-agent systems,” in *Proceedings of the 45th IEEE Conference on Decision and Control*, December 2006, pp. 3628-3633.

[6] M. Zavlanos, M. Egerstedt, and G. Pappas, “Graph theoretic connectivity control of mobile robot networks,” *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1525-1540, 2011.

[7] H. Cotuk, K. Bicakci, B. Tavli, and E. Uzun, “The impact of transmission power control strategies on lifetime of wireless sensor networks,” *IEEE Transactions on Computers*, vol. 63, no. 11, pp. 2866-2879, 2014.

[8] P. Gupta, and P. R. Kumar, “Critical power for asymptotic connectivity in wireless networks,” in *Stochastic Analysis, Control, Optimization and Applications*, Boston: Birkhuser, 1999, pp. 547-566.

[9] J. Gomez, and A. T. Campbell, “Variable-range transmission power control in wireless ad hoc networks,” *IEEE Transactions on Mobile Computing*, vol. 6, no. 1, pp. 87-99, 2007.

[10] R. Ramanathan, and R. Rosales-Hain, “Topology control of multihop wireless networks using transmit power adjustment,” in *Proceedings of the 19th Annual Joint*

Conference of the IEEE Computer and Communications Societies, March 2000, pp. 404-413.

[11] M. Kubisch, H. Karl, A. Wolisz, L. C. Zhong, and J. Rabaey, "Distributed algorithms for transmission power control in wireless sensor networks," in *IEEE Wireless Communications and Networking Conference*, March 2003, pp. 558-563.

[12] T. ElGbatt, and A. Ephremides, "Joint scheduling and power control for wireless ad hoc networks," *IEEE Transactions on Wireless Communications*, vol. 3, no. 1, pp.74-85, 2004.

[13] A. V. Fiasco, and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. Society for Industrial and Applied Mathematics, 1990.

[14] S. Boyd, and L. Vandenberghe, *Convex Optimization*. 1st ed. New York: Cambridge University Press, 2004.

[15] M. M. Asadi, H. Mahboubi, J. Habibi, A. G. Aghdam, and S. Blouin, "Connectivity assessment of random directed graphs with application to underwater sensor networks," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 4, pp. 1457 - 1464, 2017.

[16] M. M. Asadi, M. Khosravi, A. G. Aghdam, and S. Blouin, "Generalized algebraic connectivity of asymmetric networks: analysis and distributed estimation," *IEEE Transactions on Automatic Control*, 2017, in-press.

[17] S. Pourazarm, and C. G. Cassandras, "Energy-based lifetime maximization and security of wireless-sensor networks with general nonideal battery models," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 2, pp. 323-335, 2017.

[18] M. Murad, A. A. Sheikh, M. A. Manzoor, E. Feleman, and S. Qaisar, "A survey on current underwater acoustic sensor network applications," *International Journal of Computer Theory and Engineering*, vol. 7, no. 1, pp. 51-56, 2015.

[19] C. G. Cassandras, T. Wang, and S. Pourazarm, "Optimal routing and energy allocation for lifetime maximization of wireless sensor networks with nonideal batteries," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 1, pp. 86-98, 2014.

[20] J. Heidemann, M. Stojanovic, and M. Zorzi, "Underwater sensor networks: applications, advances and challenges," *Philosophical Transactions of the Royal Society A*, vol. 370, pp. 158-175, 2012.

[21] J. H. Chang, and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," *IEEE Transactions on Networking*, vol. 12, no. 4, pp. 609-619, 2004.

[22] H. Mahboubi, W. Masoudimansour, A. G. Aghdam, and K. Sayrafian-Pour, "Maximum lifetime strategy for target monitoring with controlled node mobility in sensor

networks with obstacles," *IEEE Transactions on Automatic Control*, vol. 61, no. 11, pp. 3493-3508, 2016.

[23] O. Cayirpunar, E. Kadioglu-Urtis, and B. Tavli, "Optimal base station mobility patterns for wireless sensor network lifetime maximization," *IEEE Sensors Journal*, vol. 15, no. 11, pp. 6592-6603, 2015.

[24] M. E. Keshin, İ. K. Altinel, N. Aras, and C. Ersoy, "Wireless sensor network lifetime maximization by optimal sensor deployment, activity scheduling, data routing and sink mobility," *Ad hoc Networks*, vol. 17, pp. 18-36, 2014.

[25] H. Yetgin, K. T. K. Cheung, M. El-Hajjar, and L. Hanzo, "A survey of network lifetime maximization techniques in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 828-854, 2017.

[26] R. M. Curry, and J. C. Smith, "A survey of optimization algorithms for wireless sensor network lifetime maximization," *Computers & Industrial Engineering*, vol. 16, pp. 145-166, 2016.

[27] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, vol. 23, no. 2, pp. 298-305, 1973.

[28] M. Esmaeilpour, A. G. Aghdam, and S. Blouin, "Joint Transmission Power Optimization and Connectivity Control in Asymmetric Networks," in *Proceedings of the 2018 American Control Conference*, June 2018, to appear.

[29] M. Esmaeilpour, A. G. Aghdam, and S. Blouin, "Lifetime Optimization and Generalized Algebraic Connectivity Control in Asymmetric Networks," submitted for conference publication.

[30] M. Esmaeilpour, A. G. Aghdam, and S. Blouin, "Connectivity, transmission power, and lifetime optimization in asymmetric networks: A distributed approach," submitted for journal publication.

[31] A. Ghosh, and S. Boyd, "Growing well-connected graphs," in *Proceedings of the 45th IEEE Conference on Decision and Control*, December 2006, pp. 6605-6611.

[32] A. Sehgal, C. David, and J. Schonwalder, "Energy consumption analysis of underwater acoustic sensor networks," in *Proceedings of the OCEANS '11 MTS/IEEE Kona Conference*, September 2011.

[33] T. Themistoklis, M. G. Rabbat, M. Johansson, and C. N. Hadjicostis, "Distributed finite-time computation of digraph parameters: left-eigenvector, out-degree and spectrum," *IEEE Transactions on Control of Network Systems*, vol. 3, no. 2, pp. 137-148, 2016.

[34] M. Ji, and M. Egerstedt, "Observability and estimation in distributed sensor networks," in *Proceedigns of the 46th IEEE Conference on Decision and Control*, December 2007, pp. 4221-4226.

[35] S. Sundaram, and C. N. Hadjicostis, "Finite-time distributed consensus in graphs with time-invariant topologies," in *Proceedings of the 2007 American Control Conference*, July 2007, pp. 711-716.

[36] M. Stojanovic, "On the relationship between capacity and distance in an underwater acoustic communication channel," in *Proceedings of the 1st ACM international workshop on Underwater networks (WUWNet)*, September 2006, pp. 41-47.

[37] R. Aragues, G. Shi, D. V. Dimarogonas, C. Sags, K. H. Johansson, and Y. Mezouar, "Distributed algebraic connectivity estimation for undirected graphs with upper and lower bounds," *Automatica*, vol. 50, no.12, pp. 3253-3259, 2014.

[38] C. Li and Z. Qu, "Distributed estimation of algebraic connectivity of directed networks," *Systems & Control Letters*, vol. 62, no. 6, pp. 517524, 2013.

[39] M. M. Asadi, M. Khosravi, A. G. Aghdam, and S. Blouin, "Generalized algebraic connectivity for asymmetric networks," in *Proceedings of the 2016 American Control Conference*, July 2016, pp. 55315536.

[40] X. Liu, "An optimal-distance-based transmission strategy for lifetime maximization of wireless sensor networks," *IEEE Sensors Journal*, vol. 15, no. 6, pp. 3484-3491, 2015.

[41] H. Yetgin, K. T. K. Cheung, M. El-Hajjar, and L. Hanzo, "Cross-layer network lifetime maximization in interference-limited WSNs," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 8, pp. 3795-3803, 2015.

[42] D. P. Bertsekas. *Nonlinear Programming*. 2nd ed. Belmost: Athena Scientific, 1999.

[43] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks*, vol. 38, no. 4, pp. 393422, 2002.

[44] S. Martinez, "Distributed interpolation schemes for field estimation by mobile sensor networks," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 491500, 2010.

[45] M. Mesbahi, and M. Egerstedt. *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[46] Y. Gao, and Z. Qin, "On computing the edge-connectivity of an uncertain graph," *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 4, pp. 981-991, 2015.

[47] N. W. Chang, C. Y. Tsai, and S. Y. Hsieh, "On 3-extra connectivity and 3-extra edge connectivity of folded hypercubes, *IEEE Transactions on Computers*, vol. 63, no. 6, pp. 1594-1600, 2014.

[48] X. Li, Y. Mao, and Y. Sun, "On the generalized (edge-) connectivity of graphs," *AUSTRALASIAN JOURNAL OF COMBINATORICS*, vol. 58, no. 2, pp. 304-319, 2014.

[49] P. Ali, S. Mukwembi, and S. Munyira, "Degree distance and vertex-connectivity," *Discrete Applied Mathematics*, vol. 161, no. 18, pp. 2802-2811, 2013.

[50] M. M. Zavlanos, M. B. Egerstedt, and G. J. Pappas, "Graph-theoretic connectivity control of mobile robot networks," *Proceedings of the IEEE*, vol 99, no. 9, pp. 1525-1540, 2011.

[51] R. Aragues, G. Shi, D. V. Dimarogonas, C. Sagues, and K. H. Johansson, "Distributed algebraic connectivity estimation for adaptive event-triggered consensus," in *Proceedings of the 2012 American Control Conference*, June 2012, pp. 32-37.

[52] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 89-103, 2005.

[53] G. Ye, H. Zhang, H. Liu, J. Cheng, and V. C. M. Leung, "Energy efficient joint user association and power allocation in a two-tier heterogeneous nNetwork," in *Proceedings of the 2016 IEEE Global Communications Conference*, December 2016, pp. 1-5.

[54] A. K. Das, R. J. Marks, M. El-Sharkawi, P. Arabshahi, and A. Gray, "Minimum power broadcast trees for wireless networks: integer programming formulations," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, April 2003, pp. 1001-1010.

[55] S. Blouin, "Intermission-based adaptive structure estimation of wireless underwater networks," in *Proceedings of the 10th IEEE Internaitonal Conference on Networking, Sensing and Control (ICNSC)*, April 2013, pp. 130-135.